

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Interface pour courrier électronique

Kukosama-Luzolo, Vincent

Award date:
1990

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE DAME DE LA PAIX.
NAMUR
INSTITUT D'INFORMATIQUE

INTERFACE POUR
COURRIER
ELECTRONIQUE

Vincent KUKOSAMA-LUZOLO

Facultés Universitaires Notre Dame de la Paix
Institut d'Informatique
Rue de Bruxelles 61, B-5000 Namur
Tél. 081-72.41.11 Télex 59.222 facnam-b Téléfax 081-23.03.91

INTERFACE POUR COURRIER ELECTRONIQUE

Résumé.

Présentation succincte d'éléments de télé-informatique intervenant dans le Courrier Electronique.

Présentation des principes et concepts d'ergonomie et de psychologie cognitive, permettant de dégager des critères de conception d'interfaces conviviales.

Sur base de ces critères, des recommandations sont formulées pour fournir à une station de travail, une interface conviviale pour l'utilisation du logiciel de courrier électronique EAN.

Un outil d'interaction sous Unix (X-Window) est proposé et des spécifications globales puis détaillées sont définies, pour la construction de l'interface.

Abstract.

Survey of teleinformatics concepts relative to Computer Based Message Systems and particularly to Electronic Mail.

Introduction to ergonomics and cognitive psychology principles providing criteria for better human-computer interactives.

Recommendations are then made and used for designing the user interface for an Unix system based Work Station. That interface uses graphical features and screen management provided by X-Window system, to enhance the presentation.

Mémoire de Licence et Maîtrise en Informatique
Septembre 1990 Institut d'Informatique
Promoteur: Professeur Ph. Van Bastelaer

Sincères remerciements à Botshula Richard
Anthony et Delvaux Alain, pour leurs
contributions dans la correction de ce texte.

Grand merci à Masuy Sonia pour son
assistance matérielle.

Très grand merci à Monsieur le Professeur
Van Bastelaer Philippe, pour ces nombreux avis et
conseils sur la forme et le fond de cet ouvrage.

Table des matières

I. COURRIER ELECTRONIQUE

I.1 Définition du Courrier Electronique.

I.2 Avantages et inconvénients du Courrier Electronique.

I.2.1 Avantages.

I.2.1.1 Avantages par rapport au téléphone.

I.2.1.2 Avantages par rapport au courrier postal.

I.2.2 Inconvénients.

I.2.2.1 Inconvénients par rapport au téléphone.

I.2.2.2 Inconvénients par rapport au courrier postal.

II. INTERFACE UTILISATEUR-ORDINATEUR

II.1 Généralités.

II.2 Application interactive: point de vue de l'utilisateur.

II.2.1 Introduction.

II.2.2 Recommandations ergonomiques concernant les paramètres de l'interface.

II.2.2.1 Séquencement des opérations.

II.2.2.2 Langage d'interaction.

II.2.2.3 Entrées et dispositifs d'entrée.

II.2.2.4 Sorties et dispositifs de présentation.

II.2.2.5 Traitement des erreurs.

II.2.2.6 Guidage.

II.2.2.7 Temps de réponse.

III. DIAGNOSTIC D'UN LOGICIEL INTERACTIF. ETUDE DE CAS EAN.

III.1 Généralités.

III.2 Fonctionnalités du logiciel.

III.3 Analyse des écrans.

III.3.1 Ecran d'affichage du fichier profile.

III.3.2 Ecrans de composition de message.

III.3.3 Ecrans d'affichage et de réception de messages.

III.4 Analyse des enchaînements d'écrans.

III.5 Synthèse des critiques formulées.

III.5.1 A propos du séquençement de commandes.

III.5.2 A propos du langage d'interaction.

III.5.3 A propos des entrées et des dispositifs d'entrée.

III.5.4 A propos des sorties et des dispositifs de
présentation

III.5.5 A propos du traitement des erreurs.

III.5.6 A propos du guidage.

III.5.7 A propos du temps de réponse.

IV. SPECIFICATION D'UNE INTERFACE CONVIVIALE POUR EAN.

IV.1 Introduction.

IV.2 Exemple de l'interface de MacEAN.

IV.2.1 Idées générales.

IV.2.2 MacEAN et la manipulation directe.

IV.2.3 Avantages de l'interface de MacEAN.

IV.2.4 Inconvénients de l'interface de MacEAN.

IV.3 X-Window: un outil de maquettage d'écran.

IV.3.1 Introduction.

IV.3.2 Concepts généraux.

IV.3.3 Principale ressource: la fenêtre.

IV.3.4 Différences entre X-Window et le système Apple
MacIntosh

IV.4 Présentation générale des écrans.

IV.4.1 Hiérarchie et présentation des commandes.

IV.4.2 Boutons d'actions.

IV.4.3 Boîte de dialogue.

IV.4.4 Critiques et justification de la présentation des
écrans et des dispositifs d'interaction.

- IV.5 Structure hiérarchique de l'interface utilisateur.
 - IV.5.1 Introduction.
 - IV.5.2 Choix de la structure.
 - IV.5.3 Justification du choix.
 - IV.5.4 Schéma et description de la structure.
 - IV.5.5 Remarques.

- IV.6 Structure modulaire de l'interface utilisateur.
 - IV.6.1 Introduction.
 - IV.6.2 Justification de la modularisation.
 - IV.6.3 Spécifications externes.
 - IV.6.4 Spécifications internes.

V. CRITIQUES DU TRAVAIL EFFECTUE ET CONCLUSIONS.

Préambule

L'objectif de cet ouvrage est de spécifier une interface conviviale pour le logiciel de courrier électronique EAN. Cette interface doit respecter des recommandations ergonomiques afin d'être le plus proche possible des attentes de l'utilisateur.

Le courrier électronique est une application qui connaît aujourd'hui un grand essor. Il compte parmi ses utilisateurs de plus en plus nombreux, beaucoup de non informaticiens peu habitués à l'usage de l'ordinateur. C'est d'eux qu'il faut en priorité se préoccuper afin d'optimiser l'interaction entre l'homme et la machine.

Pour optimiser ce dialogue, nous pensons qu'il est essentiel de reporter vers la machine, une partie de l'effort cognitif demandé à l'utilisateur. C'est pourquoi en tant qu'analyste, nous nous attacherons à intégrer le point de vue de l'utilisateur dans la conception de notre interface.

Partant de ces bases, nous nous intéresserons plus particulièrement à l'interface du logiciel de courrier électronique EAN. Nous en soulignerons les faiblesses en nous appuyant sur des recommandations ergonomiques, nous proposerons ensuite des spécifications pour une interface plus conviviale et nous finirons par confronter nos propres spécifications aux recommandations ergonomiques.

Notre ouvrage se subdivise en cinq chapitres. Le premier chapitre: "*Courrier électronique*" définit formellement ce que nous entendons par courrier électronique avant d'en donner les avantages et les inconvénients.

Le second chapitre: "*Interface utilisateur-ordinateur*" présente, principalement sous forme de recommandations, les pratiques qui nous serviront de guide aussi bien dans le diagnostic d'un logiciel interactif existant, que dans la conception d'une nouvelle interface.

Sur base de ces recommandations, le troisième chapitre intitulé "*Diagnostic d'un logiciel interactif. Etude de cas EAN*" établit un diagnostic des possibilités de dialogue offertes par EAN. Ces possibilités de dialogue nous les déduisons à partir des copies d'écran obtenues par notre propre expérimentation du logiciel.

Dans le quatrième chapitre intitulé "*Spécification d'une interface conviviale pour EAN*", nous tirons des enseignements des critiques formulées dans le chapitre précédent et nous proposons en conséquence une interface plus conviviale.

Avant d'en arriver aux spécifications proprement dites, ce chapitre montre en exemple le logiciel MacEAN pour illustrer l'intérêt ergonomique de la manipulation directe dans les applications interactives. Il présente aussi X-Window, un logiciel de gestion d'écran sous Unix.

Des dessins d'écrans réalisables avec X-Window sont ensuite spécifiés, ils représentent l'interface EAN que nous proposons à l'utilisateur.

Nous critiquons et justifions cette présentation des écrans puis finalement nous présentons plus formellement la structure hiérarchique et la structure modulaire de l'interface utilisateur.

Le cinquième chapitre: "Critiques du travail effectué et conclusions" fait une rétrospective de l'ensemble de notre ouvrage et en tire les conclusions.

I. COURRIER ELECTRONIQUE

Dans ce chapitre introductif, nous proposons la définition du courrier électronique adoptée par EMA (Electronic Mail Association).

Nous présentons ensuite les avantages et les désavantages de ce "progrès technologique" que constitue le courrier électronique, par rapport aux modes de communications dits traditionnels que sont le courrier postal et le téléphone.

I.1 DEFINITION DU COURRIER ELECTRONIQUE

Nous reprenons la définition adoptée par EMA et citée par [1] :

Electronic Mail is the generic name for non interactive communication of text, data, image or voice message between a sender and designated recipients by systems utilizing communication links.

[1] qui en décrit les propriétés considère chacune des technologies suivantes comme étant du courrier électronique: le telex, le facsimilé, le teletex et les systèmes de messagerie informatisés.

En effet, le telex est un moyen de communication non interactif de texte, entre un terminal expéditeur et un terminal récepteur qui stocke le message sur papier, disquette ou bande magnétique, avant de le remettre au destinataire.

Le facsimilé est aussi non interactif, le message à transmettre n'est ni une voix ni un texte, mais une image. L'image est lue par un scanner puis réduite par une technique de compression digitale, avant d'être transmise par ligne téléphonique.

Le teletex permet la communication non interactive de documents entre deux machines. Formellement, le CCITT (Comité Consultatif International de Télégraphie et Téléphonie) en donne la définition suivante tirée de [1]:

Teletex is an international service offered by administrations or recognized private operating agencies (RPOAs), enabling subscribers to exchange correspondence on an automatic memory-to-memory basis via telecommunications networks.

I.2 AVANTAGES ET INCONVENIENTS DU COURRIER ELECTRONIQUE

I.2.1 AVANTAGES

Chacun des modes de communication traditionnels (téléphone, poste, ...) présente à des degrés divers ses propres avantages. Nous soulignerons dans ce paragraphe les avantages que présente le courrier électronique par rapport au téléphone d'une part et au courrier postal d'autre part.

I.2.1.1 Avantages par rapport au téléphone

Le courrier électronique est non interactif, de ce fait il a l'avantage sur le téléphone de ne pas requérir la présence du destinataire pour pouvoir communiquer.

De plus, il donne la possibilité de composer d'abord un brouillon de message qu'on peut relire et éventuellement modifier avant d'envoyer la version finale. L'expéditeur et le destinataire gardent une trace du message, ce que ne permet généralement pas le téléphone. Le courrier électronique permet aussi d'envoyer du texte et même des images, ce que ne peut le téléphone.

Signalons aussi que le courrier électronique permet une amélioration de la correspondance, notamment quand la différence des fuseaux horaires rend les communications téléphoniques gênantes.

I.2.1.2 Avantages par rapport au courrier postal

Contrairement au courrier postal, dans un système de courrier électronique, un même message peut être diffusé à plusieurs destinataires sans qu'il ne soit nécessaire de rédiger autant d'exemplaires du message qu'il y a de destinataires. Il faut distinguer ici, trois cas de figures possibles:

- envoyer un même message à plusieurs personnes qui ne sont pas caractérisées par un nom commun de groupe et qui sont géographiquement dispersées.

Il suffit de rédiger un seul exemplaire du message et d'indiquer dans son en-tête, les adresses de tous les destinataires.

Prenons par exemple le cas où un usager Français désire lancer la même invitation à trois de ses amis se trouvant respectivement à Singapour, San Diego et Melbourne. Il composera un seul message dont l'en-tête contiendra les adresses des trois amis.

- envoyer un même message à plusieurs personnes qui ne sont pas identifiées par un nom commun de groupe, et qui sont géographiquement regroupées.

Il suffit de rédiger un seul exemplaire du message et indiquer dans son en-tête, les adresses de tous les destinataires. Ici, outre l'avantage de ne composer qu'un seul message, la duplication de celui-ci peut se faire le plus tard possible et réduire ainsi les coûts d'expédition.

Si par exemple depuis Bruxelles un usager veut atteindre trois correspondants situés à Boston, Miami et Los Angeles, un seul message sera envoyé de Bruxelles vers les Etats Unis. Ce n'est qu'à partir du continent américain que le message sera reproduit en trois exemplaires qui seront transmis vers leurs villes de destination.

- envoyer un même message à un groupe de personnes identifiées par un nom commun de groupe.

Dans ce cas aussi on ne composera évidemment qu'un seul message, dont l'en-tête n'indiquera comme destinataire que le nom du groupe.

Supposons que l'on désire transmettre une même information à un groupe de chercheurs géographiquement éloignés les uns des autres, mais travaillant au sein d'une même organisation identifiée par un nom générique.

Si on utilise le courrier postal, il faut nécessairement rédiger un exemplaire de message par chercheur. Par contre le courrier électronique dans pareille situation, permet de diffuser un même exemplaire du message à l'ensemble de tous ces chercheurs identifié par le nom générique du groupe.

Il y a donc amélioration de la rapidité et de la régularité de la transmission de l'information, chaque utilisateur du système possédant sa boîte aux lettres. Par voie postale, dans chacun des exemples cités, il aurait fallu non seulement tripler le nombre de lettres à rédiger, mais aussi et surtout tripler les coûts d'expédition par rapport au cas où il n'y aurait eu qu'un seul destinataire.

Autre avantage, le système de courrier électronique est intégré au système informatique et peut donc directement bénéficier des services offerts par les systèmes de traitement de texte et les systèmes de gestion de bases de données par exemple.

Enfin, le dernier exemple ci-après dû à [1], illustre quantitativement la réduction du cycle de communication qu'apporte le courrier électronique, en l'occurrence le facsimilé communément appelé fax.

Supposons qu'une compagnie devant préparer une offre de contrat, dispose de dix jours pour le faire. De plus, admettons que cinq personnes soient impliquées dans cette préparation et qu'elles soient réparties dans les différents départements suivants: direction générale, ingénierie, ventes, marketing et conseil juridique.

Le responsable du marketing prend les premières décisions, il prépare un premier brouillon et en envoie un exemplaire à chacun des responsables des quatre autres départements. Chacun de ces derniers annote commentaires et remarques sur l'exemplaire reçu puis retourne le document ainsi révisé, au département du marketing. Suite aux amendements individuels apportés par ses collègues, le responsable du marketing prépare une version mise à jour qu'il envoie de nouveau à chacun de ses collègues, ... et ainsi de suite jusqu'à l'obtention d'un document final viable.

La table 1 ci-après proposée par [1] montre ce que serait le cycle des communications par voie postale, si cinq étapes sont nécessaires avant l'établissement du document final. Les chiffres ont été publiés en novembre 1986 par EMMS (Electronic Mail & Micro Systems). Le tarif postal est celui du Federal Express: 12.50 \$ le courrier, soit un total de 50 \$ par étape, puisque chaque brouillon du document est expédié vers quatre destinations différentes.

Table 1. Cycle de préparation d'un document.
Acheminement par voie postale.

étape	coût de la transmission	temps de travail	temps nécessaire
I. Préparation brouillon 1. Envoi.	50 \$	4 heures	1 jour
Commentaires. Retour brouillon.	50 \$	4 heures	1 jour
II. Création brouillon 2. Envoi.	50 \$	3 heures	1 jour
Commentaires. Retour brouillon.	50 \$	2 heures	1 jour
III. Création brouillon 3. Envoi.	50 \$	2 heures	1 jour
Commentaires. Retour brouillon.	50 \$	2 heures	1 jour
IV. Création brouillon 4. Envoi.	50 \$	1 heure	1 jour
Commentaires. Retour brouillon.	50 \$	1 heure	1 jour
V. Préparation dossier final Envoi.	50 \$	1 heure	1 jour
TOTAUX	450 \$	20 heures	9 jours

Cet exemple montre le temps nécessaire au premier rédacteur pour créer les brouillons, ainsi que le temps nécessaire aux quatre "réviseurs" pour annoter leurs remarques. La préparation du document nécessite au total 20 heures de travail mais s'étale sur neuf jours.

Nous reprenons ci-après le même exemple mais où pour la livraison des documents on utilise le fax plutôt que la poste. Un scanner est utilisé pour lire le document et le transmettre à la machine à une vitesse de trois pages par minute. On suppose que le document final comme chaque brouillon intermédiaire, contient dix pages et est envoyé à quatre endroits différents en 30 minutes, y compris le setup initial. Dans ces conditions [1] obtient la table 2 ci-dessous.

Table 2. Cycle de préparation d'un document.
Acheminement par fax.

étape	coût de la transmission	temps de travail	temps nécessaire
I. Préparation brouillon 1. Envoi.	15 \$	4 heures	
Commentaires. Retour brouillon.	15 \$	4 heures	1 jour
II. Création brouillon 2. Envoi.	15 \$	3 heures	
Commentaires. Retour brouillon.	15 \$	2 heures	
III. Création brouillon 3. Envoi.	15 \$	2 heures	1 jour
Commentaires. Retour brouillon.	15 \$	2 heures	
IV. Préparation brouillon 4. Envoi.	15 \$	1 heure	
Commentaires. Retour brouillon.	15 \$	1 heure	
V. Préparation dossier final Envoi.	15 \$	1 heure	1 jour
TOTAUX	135 \$	20 heures	3 jours

La comparaison de ces deux tables montre que l'utilisation du facsimilé permet, contrairement au courrier postal (même express), qu'un document fasse plus d'un aller-retour le même jour. C'est l'un des avantages majeurs de tous les systèmes de courrier électronique par rapport au courrier postal.

Limitons le courrier électronique, à l'échange informatisé de messages entre personnes possédant chacune sa boîte aux lettres électronique et transposons la comparaison que nous venons de faire, au cas de la Belgique.

Le tarif postal est celui des PTT. L'expédition d'un exemplaire du document revient à 34 F, soit 136 F pour 4 exemplaires, ce qui nous donne la table 3 ci-après.

Table 3. Cycle de préparation d'un document.
Acheminement par les PTT.

étape	coût de la transmission	temps de travail	temps nécessaire
I. Préparation brouillon 1. Envoi.	136 F	4 heures	1 jour
Commentaires. Retour brouillon.	136 F	4 heures	1 jour
II. Création brouillon 2. Envoi.	136 F	3 heures	1 jour
Commentaires. Retour brouillon.	136 F	2 heures	1 jour
III. Création brouillon 3. Envoi.	136 F	2 heures	1 jour
Commentaires. Retour brouillon.	136 F	2 heures	1 jour
IV. Création brouillon 4. Envoi.	136 F	1 heure	1 jour
Commentaires. Retour brouillon.	136 F	1 heure	1 jour
V. Préparation dossier final Envoi.	136 F	1 heure	1 jour
TOTAUX	1224 F	20 heures	9 jours

Le tarif du courrier électronique nous a été aimablement communiqué par le département *Transmission de Données* de la RTT. Il concerne le système *DCS.400* qui est une extension du système *DCS.MAIL*: le service de messagerie électronique de la RTT.

Ce tarif (TVA de 19 % incluse) comprend :

- la taxe fixe de 11,9 F par message;
- la taxe proportionnelle de 3,57 F pour 1000 caractères;
- la taxe de remise diffusée de 3,57 par destinataire au-delà du premier.

Nous ne prenons en compte ni les frais de raccordement (11.950 F), ni la redevance d'abonnement bimestrielle (5950 F).

Rappelons que chaque document transmis contient 10 pages. Nous supposons que chaque page contient 1.500 caractères, soit un total de 15.000 caractères par document. Dans ces conditions, nous obtenons approximativement la table 4 ci-après.

Table 4. Cycle de préparation d'un document.
Acheminement par DCS.400

étape	coût de la transmission	temps de travail	temps nécessaire
I. Préparation brouillon 1. Envoi.	80 F	4 heures	1 jour
Commentaires. Retour brouillon.	262 F	4 heures	
II. Création brouillon 2. Envoi.	80 F	3 heures	
Commentaires. Retour brouillon.	262 F	2 heures	
III. Création brouillon 3. Envoi.	80 F	2 heures	1 jour
Commentaires. Retour brouillon.	262 F	2 heures	
IV. Préparation brouillon 4. Envoi.	80 F	1 heure	
Commentaires. Retour brouillon.	262 F	1 heure	
V. Préparation dossier final Envoi.	80 F	1 heure	
TOTAUX	1448 F	20 heures	3 jours

Au vu de ces deux dernières tables, nous pouvons affirmer que les observations faites sur la première comparaison restent valables.

Citons enfin d'autres avantages indirects qui en découlent:

- Amélioration de l'interaction et de la prise de décision au sein d'un groupe;
- Diminution des interruptions et meilleure gestion du temps.

I.2.2 INCONVENIENTS

Lorsqu'utilisé abusivement, le Courrier Electronique présente quelques inconvénients que nous mettons en évidence dans les lignes qui suivent.

I.2.2.1 Inconvénients par rapport au téléphone

Si le téléphone supprime le contact humain direct, il garde au moins un contact vocal entre les correspondants. Il n'en est pas de même du courrier électronique qui dans les relations humaines joue un rôle négatif.

Le fait par exemple qu'on puisse diffuser un même message vers plusieurs destinataires qu'il suffit de spécifier dans l'en-tête, vise à atteindre le plus de monde à moindre effort. Dans les faits cela se traduit par des messages impersonnels, ce qui n'est pas possible avec le téléphone, qui par essence ne véhicule que des correspondances personnalisées.

Le problème général de la sécurité des systèmes informatiques se reporte aussi sur les systèmes de courrier électronique. La possibilité qu'un indiscret puisse se connecter au réseau pose non seulement un problème d'identification du correspondant, mais aussi celui de son authentification. Par correspondant nous entendons aussi bien l'émetteur que le récepteur.

I.2.2.2 Inconvénients par rapport à la poste

Reprenons l'exemple du groupe de chercheurs cité plus haut et supposons qu'il comprenne cinquante membres. Si l'on désire envoyer une même note à dix d'entre eux, il serait plus facile de diffuser cette note à l'ensemble du groupe plutôt que d'adresser individuellement une note à chacun des chercheurs concernés. Mais le problème ici c'est qu'en diffusant l'information aux cinquante chercheurs du groupe, quarante d'entre eux qui ne sont pas du tout concernés par le message, le reçoivent malgré tout.

Il est évident que dans ces conditions, chaque chercheur sera submergé de courriers inutiles et encombrants. Par conséquent, il devra régulièrement sélectionner d'abord les messages qui ne le concernent pas, les détruire, puis seulement lire ceux qui lui sont effectivement destinés (s'il en reste).

Dans son ouvrage, [1] observe d'ailleurs à ce propos que souvent l'utilisateur est vite démotivé et ne cherche même plus à prendre connaissance du flot de messages qui l'inonde. De plus, si le volume des échanges était abusivement ou accidentellement trop important, il s'en suivrait un ralentissement de la machine. Cela pénaliserait tous les utilisateurs, même ceux qui ne sont pas concernés par le système de courrier électronique.

II. INTERFACE UTILISATEUR-ORDINATEUR

Au chapitre précédent nous avons brièvement présenté le domaine d'application auquel notre travail s'intéresse plus particulièrement.

Pour une application quelconque donnée, nous allons décrire dans les paragraphes qui suivent, les critères à prendre en compte pour réaliser une interface permettant de rendre le logiciel facile à apprendre et agréable à utiliser. En d'autres termes nous allons tenter d'indiquer comment adapter le logiciel aux conditions de travail de l'utilisateur.

II.1 GENERALITES

L'objectif de ce chapitre est en fait de montrer comment intégrer des éléments ergonomiques dans la conception des interfaces utilisateur-ordinateur. Cette intégration vise à optimiser l'interaction entre l'homme et la machine.

Des principes relevant de l'ergonomie et de la psychologie cognitive permettent la conception de logiciels interactifs dont l'apprentissage et l'usage sont plus adaptés à l'utilisateur.

Cet utilisateur qui dans la majorité des cas n'est pas informaticien, pense sa tâche dans un univers psychologique qui lui est propre, tandis que l'univers dans lequel il exécute les actions pour accomplir sa tâche est l'univers physique de l'interface. Pour un complément d'informations, nous convions les personnes intéressées par cet aspect du problème, à la lecture de l'ouvrage de [10].

Prenons comme exemple un utilisateur de MS WORD qui dans son univers psychologique, pense la tâche consistant à formater un paragraphe. Les actions qu'il exécutera dans l'univers physique de l'interface sont: déplacer le curseur, sélectionner la commande de formatage, fixer les largeurs des marges et valider la commande sélectionnée.

Entre ces deux univers existe un fossé qu'il faut autant que possible réduire. Pour y parvenir, le concepteur devra créer une interface telle que l'effort cognitif demandé à l'utilisateur soit minimal. Il s'agit aussi bien de l'effort pour traduire le résultat dans le monde de la représentation mentale (intention) de l'utilisateur, que de l'effort pour traduire l'univers psychologique en univers physique.

Un logiciel interactif est conçu, évalué et utilisé par des personnes logiquement différentes, ayant des objectifs différents et des perceptions différentes de l'interaction homme-machine.

Dans le cadre de nos préoccupations, les principaux intervenants sont: l'informaticien, l'ergonome et l'utilisateur.

Nous considérons que l'objectif de l'utilisateur est d'avoir un logiciel lui permettant d'exécuter sa tâche avec le plus d'aisance possible. Une étude de [8] tient compte des paramètres suivants pour décrire la perception des utilisateurs vis-à-vis du logiciel:

- lisibilité des sorties;
- adaptation du logiciel aux situations réelles de travail;
- aide à l'utilisation et à l'apprentissage;
- initiatives possibles pour l'utilisateur.

Dans son ouvrage, [7] indique que l'ergonome observe le couple homme-machine de l'extérieur et en vue d'améliorer la communication au sein de ce couple. Il s'intéresse aux paramètres suivants:

- séquençement des opérations;
- langage d'interaction;
- entrées et dispositifs d'entrée;
- sorties et dispositifs de présentation;
- traitement des erreurs;
- guidage.
- temps de réponse.

Chacun de ces paramètres sera vu en détail au prochain chapitre.

Dans le même ouvrage nous lisons aussi que selon qu'il est analyste ou programmeur, l'informaticien a un objectif distinct. S'il est analyste, il cherche à définir les spécifications des fonctionnalités de l'application. S'il est programmeur, il cherche à mettre en oeuvre un logiciel fiable, respectant les spécifications fonctionnelles sous les contraintes du matériel utilisé.

Nous n'avons pu recueillir sur ces différents points, ni les avis des utilisateurs, ni ceux des informaticiens, ni ceux des ergonomes. Nous pensons cependant que les paramètres dont tient compte l'étude de [8] sont si généraux qu'effectivement, selon notre propre expérience d'utilisateur, ils suffisent pour décrire l'ensemble des perceptions des utilisateurs de logiciels interactifs.

Bien que nous partageons l'avis de [7] sur les objectifs du programmeur et ceux de l'analyste, nous noterons toutefois que de notre expérience, lorsque pour un même logiciel ou une même partie de logiciel, la même personne est à la fois analyste et programmeur, la séparation et l'indépendance de l'un vis-à-vis de l'autre n'est plus aussi nette et les objectifs de l'un sont alors conditionnés par les objectifs de l'autre.

II.2 APPLICATION INTERACTIVE: POINT DE VUE DE L'UTILISATEUR.

II.2.1 INTRODUCTION

De notre expérience, il ressort que l'utilisateur qui sur une longue période répète souvent les mêmes tâches, finit par acquérir des automatismes qui lui permettent d'exécuter ces tâches très rapidement et avec une mobilisation minimale de ses processus conscients. De ce fait, la conception d'une interface doit être telle qu'elle favorise l'acquisition de ces automatismes.

Cela ne pourra se produire que si dans toutes les situations identiques, l'action de l'utilisateur est toujours la même.

II.2.2 RECOMMANDATIONS DE L'ERGONOMIE CONCERNANT LES PARAMETRES DE L'INTERFACE

Les recommandations qui suivent prennent en compte les sept paramètres relevés par [7]: le séquençement des opérations, le langage d'interaction, les entrées et dispositifs d'entrée, les sorties et dispositifs de présentation, le traitement des erreurs, le guidage et le temps de réponse.

II.2.2.1 Séquençement des opérations

Le séquençement des opérations est selon [7], une adéquation entre l'ordre des opérations fixé par l'ordinateur et l'ordre dans lequel l'utilisateur pense sa tâche.

Une conception d'interface basée, comme c'est souvent le cas, sur le déroulement normal de la tâche, ne tient compte ni de l'existence de différents types d'utilisateurs, ni des aléas de l'environnement. Ces deux éléments constituent pourtant le cadre naturel de toute activité humaine.

Le séquençement des opérations induit des séquences de commandes dont nous présentons quelques recommandations proposées par [6].

- Les séquences de commandes doivent être simplifiées afin de permettre à l'utilisateur d'avoir des actions rapides et d'exécuter le plus facilement possible, toutes les possibilités qui lui sont offertes.

- Les facilités des séquences de fonctions offertes à l'utilisateur doivent correspondre à ses objectifs. Par exemple, les actions fréquentes doivent être faciles à exécuter tandis que celles aux effets destructeurs et irréversibles, suffisamment difficiles pour nécessiter une plus grande attention de la part de l'utilisateur.

- L'utilisateur doit pouvoir adapter les séquences des opérations en fonction de son expérience. Rythme lent pour l'utilisateur inexpérimenté et cadence rapide pour l'expérimenté, grâce à des raccourcis ou des abréviations par exemple.

- Pour des commandes fréquemment utilisées, l'utilisateur doit aussi avoir la possibilité de se définir ses propres macro-commandes.

- C'est l'utilisateur qui doit contrôler les séquences de commandes et non les traitements internes du logiciel.

- Si les environnements matériel et logiciel le permettent, les options suivantes seront définies:

- option de retour arrière permettant de revenir à l'affichage obtenu lors de la transaction précédente;

- option de recommencement ayant pour effet de revenir au premier affichage d'une séquence de transactions. Cette option donne à l'utilisateur la possibilité de revoir une transaction d'entrées située en début de séquence, et d'effectuer le cas échéant, les changements nécessaires;

- option de destruction ayant pour effet d'annuler toutes les entrées dans une séquence de transactions;

- option de fin ayant pour effet de conclure une séquence de transactions et de retourner au menu général des options.

- Lorsque l'utilisateur exécute une commande, les effets qui s'en suivent doivent lui être signalés sans retard et sans ambiguïté, soit par la production immédiate du résultat attendu, soit par un message indiquant que l'exécution est en cours ou différée, ou que la commande nécessite une correction ou une confirmation.

Ce retour d'information doit rassurer l'utilisateur sur l'état fonctionnel du système, lui éviter l'anxiété due à un temps de réponse trop long (dépassant une valeur de l'ordre de 4 secondes), l'informer sur la progression dans l'exécution des fonctions de l'application et l'informer sur l'espace de travail (par exemple: bien localiser la fenêtre courante, la fenêtre de diagnostic, ...)

Pour commenter ces recommandations nous dirons que globalement, il est nécessaire de les respecter si l'on veut obtenir une interface de bonne qualité, c'est-à-dire permettant à l'utilisateur d'exécuter toutes ses tâches avec facilité.

La simplicité et la rapidité d'exécution des commandes constituent à notre avis la caractéristique la plus importante à retenir à ce niveau-ci. Elle permet non seulement de réduire le temps d'apprentissage mais aussi de conserver plus longtemps les connaissances acquises.

II.2.2.2 Langage d'interaction

Le langage d'interaction est selon [7], l'ensemble des conventions qui permettent à l'utilisateur d'exprimer au moyen d'un vocabulaire et d'une syntaxe, les opérations qu'il désire faire effectuer par la machine. Ces opérations portent sur des données et/ou des commandes.

II.2.2.2.1 Vocabulaire et syntaxe

(i) Le vocabulaire.

Il sera emprunté au langage des spécialistes de la tâche plutôt qu'au langage des informaticiens. Une étude faite par [8] montre en effet que l'informaticien pense en fonction de la machine et de sa culture informatique et de ce fait, utilise soit des termes qui ne reflètent que son point de vue et non celui de l'utilisateur, soit des termes qui ont un sens différent selon qu'on est dans le contexte de l'informatique ou non.

L'informaticien parlera par exemple "d'afficher" un menu alors que l'utilisateur pensera à "consulter" un menu.

(ii) La syntaxe.

La syntaxe regroupe toutes les règles permettant d'exprimer et de combiner entre elles les commandes et les données. Deux caractéristiques sont essentielles pour que la manipulation du logiciel soit d'un apprentissage et d'un usage faciles: - la simplicité pour des raisons de mémorisation et - l'homogénéité pour diminuer les risques d'erreurs et favoriser l'acquisition d'automatismes.

II.2.2.2.2 Types de dialogue

Nous entendons par dialogue, une suite finie d'échanges d'informations entre un opérateur humain (l'utilisateur), et une application interactive. Les différents types de dialogues que nous distinguons sont: le dialogue par questions/réponses, le dialogue par remplissage de formulaire, le dialogue par menus, le dialogue par touches fonctions et le dialogue par langage de commandes.

Nous présentons ci-après chacun de ces types de dialogue, mais le lecteur pourra aussi consulter l'ouvrage de [5] pour des informations complémentaires.

(i) Question/réponse.

Le dialogue par question/réponse est entièrement piloté par l'ordinateur qui pose des questions auxquelles l'utilisateur répond sans possibilité de prendre l'initiative.

- Ce mode ne sera recommandé que pour les situations où les informations à entrer ne peuvent être contenues dans une liste ou sont difficiles à encoder. Il permet d'avoir des taux d'erreurs relativement bas, mais il est par contre lourd pour l'utilisateur expérimenté.

(ii) Remplissage de formulaire.

Plus rapide que le mode précédent, le remplissage de formulaire reste cependant entièrement piloté par le logiciel. C'est lui qui à chaque étape présente un formulaire que l'utilisateur complète d'un seul coup.

- Ce mode est souhaitable lorsqu'une flexibilité est nécessaire, pour l'entrée d'éléments optionnels notamment. Si l'utilisateur doit entrer des informations déjà imprimées sur un formulaire en papier, il est utile que l'écran soit autant que possible isomorphe à ce formulaire.

(iii) Menus.

Les menus conviennent aussi bien à la sélection des commandes qu'au choix des arguments des commandes et sont d'un usage pratique, si des dispositifs de pointage direct (souris, crayon optique,...) sont utilisés.

L'utilisation des menus est conseillée quand on souhaite limiter les erreurs lors des premières manipulations. De plus, ils constituent un aide-mémoire pour l'utilisateur quand le nombre de commandes est si élevé qu'il dépasse ses capacités de mémorisation.

[18] et [19] traitent largement le problème des menus dans les interfaces homme-ordinateur et voici quelques recommandations empiriques formulées par [6].

- Pour les usagers inexpérimentés, chaque menu ne devra requérir qu'une seule sélection à la fois, laquelle se fera par pointage direct;

- Les options du menu doivent être formulées d'une façon permettant le pointage direct ou l'entrée d'un code, plutôt qu'exprimées sous forme de questions.

Intuitivement nous dirons qu'il y a pointage direct ou manipulation directe, lorsqu'un opérateur humain manipule et guide un dispositif relié à un ordinateur, qui en enregistre toutes les actions et en répercute les mouvements sous forme de déplacement de curseur ou pointeur à l'écran;

- Si l'entrée se fait par des codes, chacun d'eux doit être la lettre initiale ou les lettres initiales de l'option qu'il représente. Ce code peut cependant être un nombre s'il y a un ordre implicite dans la liste des options. Les codes doivent être utilisés de façon homogène et les options correspondantes doivent être localisées de façon homogène aussi dans toutes les étapes du dialogue.

- En cas de numérotation, celle-ci commencera par le nombre 1 et chaque nombre sera suivi d'un point et d'un espace blanc.

- Les options doivent être ordonnées suivant un ordre logique par rapport à la tâche à accomplir, ou suivant leur fréquence d'utilisation.

- Un menu ne doit afficher que des options appropriées à l'étape de la transaction dans laquelle se trouve l'utilisateur.

(iv) *Touches fonctions.*

Le dialogue par touches fonctions est généralement à l'initiative de l'utilisateur qui presse une touche pour indiquer l'action désirée.

Les touches fonctions ont l'avantage de réduire le nombre de frappes et de minimiser ainsi les erreurs typographiques.

Par contre, leur principal inconvénient est que lorsque le nombre des opérations possibles nécessite de nombreuses touches fonctions, il faut disposer d'un clavier volumineux et coûteux qui en outre, ne permet pas de passer facilement à une autre application.

Ce mode de dialogue reste toutefois bien approprié pour les utilisateurs débutants, mais aussi pour les plus expérimentés lorsque la tâche nécessite l'entrée rapide et sans erreur de commandes critiques et fréquentes.

Voici les recommandations les plus pertinentes que nous retiendrons de l'ouvrage de [6]:

- Ne pas assigner la même touche à plusieurs fonctions, même dans des contextes différents. Cette solution supprimerait certes la nécessité de disposer de claviers volumineux, mais elle est génératrice d'erreurs car l'utilisateur peut oublier le contexte d'utilisation de la touche.

- Les fonctions qui ne sont pas nécessaires dans une transaction particulière doivent être désactivées, c'est-à-dire qu'elles ne doivent donner lieu à aucune action si les touches correspondantes sont sélectionnées.

- Les touches fonctions doivent nécessiter une seule pression pour être activées.

(v) *Langages de commandes*

Les langages de commandes sont un type de dialogue à l'initiative de l'utilisateur. Une ligne de commande (ou commande tout simplement) est constituée d'un terme de commande qui définit la fonction, et de paramètres ou arguments qui spécifient les différentes options de la fonction.

Plus que tout autre mode, les langages de commandes autorisent beaucoup de flexibilité. Quand il faut par exemple intégrer de nouvelles fonctions dans l'application, ou même quand des fonctions existantes vont être utilisées plus fréquemment qu'avant, les changements à apporter sont mineurs et ne concernent que le logiciel, pas le matériel.

Les inconvénients sont cependant nombreux et parmi tant d'autres, citons les plus flagrants relevés par [6]:

- L'apprentissage de l'ensemble des commandes est difficile, d'autant plus que l'utilisateur n'emploie de manière régulière qu'un sous-ensemble restreint d'entre elles. Il doit de ce fait systématiquement recourir au manuel d'utilisation ou dans les meilleurs des cas, à une aide en ligne pour déterminer la syntaxe, la dénomination et les effets des commandes les moins utilisées.

- La nécessité de frapper plusieurs touches pour composer la commande accroît le risque d'erreurs typographiques et ralentit la vitesse d'exécution du travail.

Ces inconvénients sont aggravés si l'utilisateur est peu habitué à l'usage du clavier, mais si toutefois on adopte ce mode de dialogue, on respectera les recommandations suivantes.

- Le langage de commandes doit être organisé en groupe de commandes fonctionnellement similaires et à différents niveaux de complexité pour tenir compte des niveaux d'expérience des usagers.

- L'utilisateur doit pouvoir obtenir du logiciel une aide lui indiquant la syntaxe exacte et les effets d'une commande donnée.

- Le nombre de commandes doit être aussi réduit que possible et la correspondance de l'ensemble des fonctions vers l'ensemble des commandes doit être univoque.

- L'écriture de la commande doit être indépendante des considérations relatives à l'architecture physique du logiciel. L'utilisateur doit par exemple pouvoir demander un fichier par son nom logique, sans fournir d'information supplémentaire relative à sa localisation physique.

- Aucune ponctuation ne devrait figurer dans la ligne de commande. En cas de nécessité absolue, un symbole standard devra être utilisé de façon homogène.

- Une ligne de commande doit être valide quelle que soit le nombre non nul d'espaces blancs avant et après chaque paramètre.

Dans leur ensemble, toutes ces recommandations sur le langage d'interaction nous paraissent aussi évidentes qu'indispensables, elles n'appelleront aucun commentaire de notre part, à l'exception toutefois du type de dialogue par menus.

Il est vrai que les menus sont très attrayants pour l'utilisateur, surtout parce qu'ils permettent de simplifier l'apprentissage d'un logiciel et de diminuer l'effort de mémorisation, comparativement au langage des commandes notamment.

Nous ferons remarquer cependant qu'à notre avis les menus ne simplifient l'apprentissage d'un logiciel que si d'une part ils sont bien structurés et que d'autre part l'utilisateur est familier avec la sélection par menu. Nous ne pensons pas en effet, qu'un utilisateur ayant une longue expérience du langage de commandes, soit nécessairement à l'aise dans l'apprentissage d'un nouveau logiciel, s'il découvre en même temps la sélection par menus.

II.2.2.3 Entrées et dispositifs d'entrée

Si la saisie des données ne peut se faire que par le clavier et ne pose aucun problème particulier, la saisie des commandes, elle, offre plusieurs modes:

- la frappe au clavier de la ligne de commande abrégée ou non;
- la frappe des touches fonctions;
- la frappe simultanée d'une touche spécialisée (Alt, Ctrl, Esc, ...) et d'une touche normale (alphanumérique);
- la désignation directe sur l'écran (crayon optique ou toucher du doigt);
- la désignation par positionnement du curseur et validation (touches directionnelles du clavier ou souris).

Les deux derniers modes requièrent le moins d'effort de mémorisation et seront donc préconisés pour les débutants et les utilisateurs occasionnels. Pour les informaticiens utilisant intensivement le clavier, l'association de touches spécialisées et normales s'avère souvent efficace, mais pour la grande majorité des non informaticiens, elle constitue un dispositif d'entrée complexe, difficile à mémoriser et générateur d'erreurs. Citons pour nous en convaincre, un exemple des plus frustrants pour un novice: l'éditeur emacs.

En cours d'édition d'un fichier, si l'on désire subdiviser l'écran en deux fenêtres horizontales, l'une contenant ce premier fichier et l'autre un deuxième fichier, la séquence de commandes suivante doit nécessairement précéder l'entrée du nom du deuxième fichier:

Ctrl-X 2 Ctrl-X Ctrl-F

où Ctrl-X représente la frappe simultanée des touches CONTROL et X du clavier. Outre la complexité de cette commande, il faut aussi ajouter qu'il existe dans Emacs,

plus de cent combinaisons valides du même genre, ce qui rend toute tentative de mémorisation quasiment vaine.

Présentons maintenant les recommandations concernant les entrées:

- Pour les paramètres acceptant des valeurs par défaut, celles-ci doivent apparaître à l'écran dans des champs appropriés.

- Les abréviations doivent toutes être déduites d'une même règle de construction et devraient être de même longueur.

- La fin d'une entrée de données doit se faire de manière explicite par une touche de validation clairement dénommée.

- L'utilisateur doit pouvoir revoir les données entrées et les modifier si nécessaire avant la validation.

- Toutes les entrées non confidentielles effectuées par l'utilisateur, doivent apparaître à l'écran.

- Quand la désignation de position se fait par curseur, celui-ci ne doit jamais éclipser le caractère qu'il désigne.

- Quand l'objectif principal de la tâche est la sélection d'items, le pointage direct sera préconisé.

- L'entrée des données doit être explicitement guidée par des messages d'aide et à cet effet, des labels ou titres descriptifs de champs doivent être clairement dénommés et placés de façon consistante et homogène sur l'écran. Ces labels doivent de plus être protégés de l'intervention de l'utilisateur.

- Quand des unités de mesure sont associées à une entrée, elles doivent être familières à l'utilisateur, apparaître dans le label du champ et ne doivent plus être entrées par l'usager.

- Chaque entrée doit être la plus courte possible afin de limiter les erreurs et favoriser leur détection.

- Un guidage à la demande aisément accessible, doit pouvoir indiquer à l'utilisateur ce qui est attendu de lui à chaque étape.

Nous pensons que ces recommandations sont fondamentales lors de la conception d'une interface conviviale. Nous émettons cependant quelques réserves quant à la déduction de toutes les abréviations à partir d'une même règle.

Il faut en effet que d'une part cette règle soit très simple pour en faciliter la mémorisation et que d'autre part, toutes les abréviations déduites à partir d'elle, soient distinctes les unes des autres. Ces deux points ne sont pas toujours conciliables.

II.2.2.4 Sorties et dispositifs de présentation

Afin que nous puissions concevoir des présentations d'écran qui facilitent la saisie visuelle de l'information, voici différentes recommandations proposées par [6], mais le lecteur consultera aussi l'ouvrage de [13] pour un complément d'informations.

- Tous les éléments présentés à l'écran doivent pouvoir être perçus sans effort par l'oeil humain. Il est donc impératif de s'assurer de la bonne taille des caractères, du contraste, de la luminosité, ...

- Chaque écran doit fournir des informations directement utilisables, ne nécessitant qu'un minimum d'interprétation, de traduction et de traitement de la part de l'utilisateur.

- L'utilisateur expérimenté doit avoir la possibilité d'interrompre des messages trop longs, inutiles, et redondants pour lui mais utiles pour le débutant.

- Les messages d'erreur doivent indiquer la cause de l'erreur, sa localisation précise, et fournir le moyen de correction approprié.

- Quand les données à afficher ne peuvent tenir sur une seule page d'écran, elles doivent être automatiquement subdivisées en pages d'écran séparées, chacune portant un titre descriptif qui identifie ses rapports avec les autres pages.

- Pour aider l'utilisateur dans l'exploration des informations sur l'écran, des techniques de groupement d'items seront utilisées suivant les quatre critères ci-après: ordre séquentiel, fréquence d'utilisation, fonctionnalité et importance.

- a) Le *groupement séquentiel* consiste à grouper les items dans leur ordre d'utilisation, la première information nécessaire étant affichée en haut.

- b) Le *groupement selon la fréquence d'utilisation* repose sur le principe selon lequel les items les plus fréquemment utilisés doivent être regroupés

- c) Le *groupement fonctionnel* est une possibilité qui consiste à grouper tous les items qui correspondent à un type particulier d'activité.

- d) Le *groupement par importance* peut être utilisé si certains items sont critiques et doivent être placés à un endroit où ils ne peuvent être négligés.

- Un label doit accompagner chaque groupe d'items afin de rendre leur identification rapide et aider l'utilisateur à parcourir plus facilement l'écran.

- Un label peut être utilisé pour identifier un seul item, un groupe d'items ou tout un écran.

La seule remarque que nous formulerons ici se rapporte à la subdivision des grands flux d'information en plusieurs pages d'écran.

Cette subdivision est certes nécessaire mais insuffisante en elle-même. Elle doit à notre avis être impérativement accompagnée d'une indication claire sur la manière dont l'utilisateur peut passer à la page suivante, revenir sur la page précédente ou mettre fin au défilement des pages.

II.2.2.5 Traitement des erreurs

Le traitement des erreurs consiste à la détection (rapide) et la correction de celles-ci. Nous distinguons deux types d'erreurs: celles d'exécution et celles d'intention.

Une erreur d'exécution est consécutive à une manipulation accidentelle d'une touche autre que celle désirée. Sa détection est généralement immédiate et la rectification facile.

Les erreurs d'intention sont celles dues à une interprétation erronée de la signification des commandes. Elles ne sont détectées que tardivement et leur correction nécessite une bonne connaissance du logiciel de la part de l'utilisateur.

Les recommandations que nous formulons concernent distinctement les deux aspects de la gestion des erreurs: le signalement et la correction.

(i) Signalement des erreurs.

- Pour garantir une homogénéité et optimiser ainsi la recherche des messages d'erreur, ceux-ci doivent être présentés toujours sous la même forme et toujours au même endroit.

- S'il y a assez de place et si on utilise une fenêtre en surimpression, le message d'erreur pourra apparaître à côté de l'erreur.

- Dans le cas contraire, on utilisera une zone spéciale pour les erreurs, avec des marques spéciales de signalement telles qu'un clignotement du curseur ou de la zone, un signal sonore ou une inversion vidéo.

(ii) Correction des erreurs.

- L'utilisateur doit pouvoir revenir sans peine sur l'opération où est apparue l'erreur et doit pouvoir annuler en totalité ou en partie le travail qu'il a effectué depuis lors.

- Aucune action ne devrait être irréversible afin d'éviter l'anxiété chez l'utilisateur et lui permettre de se sentir en confiance pour explorer d'autres possibilités du logiciel.

Ces recommandations nous paraissent évidentes et ne ferons l'objet d'aucun commentaire.

II.2.2.6 Guidage

Nous retiendrons les recommandations suivantes dues à [6]:

- Décrire pour chaque commande son rôle et ses effets. Ajouter à l'intention des novices, un niveau d'explication détaillée à la demande; ce niveau fournira une liste des différents problèmes que l'on peut rencontrer pour exécuter la commande.

- Prévoir pour l'utilisateur expérimenté la possibilité de se remémorer la syntaxe et la sémantique d'une commande.

Il serait effectivement souhaitable, que sur demande, l'utilisateur (novice) puisse obtenir la liste de tous les problèmes se rapportant à une commande données. Cependant, il faut remarquer que cela ne serait envisageable que si le nombre de problèmes possibles se rapportant à chaque commande est très restreint. Dans le cas contraire, cette recommandation n'a pas de sens.

Il est en effet impensable qu'un utilisateur puisse par exemple demander la liste de toutes les erreurs syntaxiques possibles, se rapportant à une commande de compilation d'un programme dans un langage donné.

II.2.2.7 Temps de réponse

Les recommandations concernant le temps de réponse se basent sur les deux observations suivantes:

- dans une conversation entre deux personnes, un délai d'environ deux secondes est un temps de réponse au-delà duquel l'interlocuteur perçoit une impression gênante d'attente;

- toute information qui entre ou qui sort du cerveau humain transite pendant un temps de l'ordre de quatre à six secondes, dans une partie de la mémoire telle que si l'intervenant est interrompu pendant la phase de mémorisation, l'information risque d'être perdue. Cette partie de la mémoire est appelée mémoire à court terme.

De ces deux observations nous déduisons qu'un temps de réponse trop long (plus de 4 secondes environ) nous oblige à garder longtemps l'information dans la mémoire à court terme, ce qui est pénible et gênant. D'où les recommandations suivantes proposées par [7]:

- Si le temps de réponse est manifestement trop long, il faut afficher un message d'attente expressif, pour que l'utilisateur puisse éventuellement faire autre chose entre temps.

- Il faut éviter des temps de réponse trop variables, afin d'éviter l'anxiété chez l'utilisateur qui ne serait pas en mesure de savoir si ce temps d'attente inhabituel provient des contraintes techniques ou d'une erreur de sa part.

- Si toutefois le temps de réponse est variable, il faut contrôler cette variabilité de sorte que des délais identiques s'écoulent entre des actions identiques de l'utilisateur et la réponse de l'ordinateur.

Le seul commentaire que nous ferons sur ces dernières recommandations c'est que le temps de réponse ne dépend pas toujours uniquement du logiciel, mais aussi des contraintes de l'environnement technique.

La gestion du temps de réponse peut donc s'avérer très hasardeuse, notamment lorsque les ressources informatiques (processeur, mémoire centrale, disques,...) sont partagées avec d'autres utilisateurs. Le temps de réponse est alors souvent variable et imprévisible.

A notre avis il est suffisant, lorsque qu'une opération nécessite un long temps de réponse, de signaler simplement qu'elle est en cours et éventuellement le rappeler à intervalle régulier, ou matérialiser sa progression de manière expressive.

III. DIAGNOSTIC D'UN LOGICIEL INTERACTIF. ETUDE DE CAS EAN

Les deux premiers chapitres nous ont respectivement introduits au courrier électronique et aux problèmes d'interface utilisateur-ordinateur.

Partant de là, tout au long de ce troisième chapitre, nous allons présenter, analyser et critiquer l'interface d'un logiciel particulier de courrier électronique: EAN.

III.1 GENERALITES

EAN est un logiciel de courrier électronique offrant à l'utilisateur les possibilités suivantes: écrire un premier brouillon de message, le sauver sur fichier, le mettre en attente, l'éditer à l'aide de l'éditeur de son choix (pour autant qu'il soit disponible) et une fois toutes les modifications faites, l'envoyer à ses destinataires avec la possibilité de demander un accusé de réception.

[4] répartit les messages que manipule EAN en deux groupes: les messages d'utilisateurs et les messages du système. Ces derniers consistent en un compte rendu ou rapport de bon déroulement ou non de la communication.

Un message d'utilisateur quant à lui comprend deux parties: un en-tête et un corps. L'en-tête contient des informations de service et d'identification de l'expéditeur et du destinataire. Le corps est le texte du message c'est-à-dire l'information proprement dite qui est transmise entre correspondants. Nous parlerons essentiellement des messages d'utilisateur dans la suite.

Du point de vue interface utilisateur, EAN propose une gestion d'écran classique, dans ce sens que l'interface ne gère qu'une seule fenêtre dans laquelle les sorties se font ligne par ligne.

Le logiciel EAN est conforme au standard X.400, celui-ci fut adopté en 1984 par le CCITT, comme norme d'interconnexion de systèmes de courrier électronique.

X.400 permet de mettre en communication des matériels de différents manufacturiers et constitue de ce fait, non seulement un standard technique, mais aussi un outil d'aide à l'implémentation du courrier électronique, puisqu'il permet d'avoir une organisation décentralisée.

Fonctionnellement X.400 définit deux groupes logiques: l'Agent d'Utilisateur ou UA pour User Agent et l'Agent de Transfert de Message ou MTA pour Message Transfer Agent.

- l'Agent d'Utilisateur s'occupe de toutes les fonctions relatives à la préparation des messages. Il s'agit par exemple de composer ou éditer un message. C'est en fait l'interface entre l'utilisateur et le système de courrier électronique;

- l'Agent de Transfert de Message coordonne l'acheminement et la diffusion des messages, en coopérant avec d'autres MTA avec lesquels il constitue un Système de Transfert de Messages (MTS pour Message Transfer System).

Le lecteur trouvera dans les ouvrages de [1], [2] et [3] de plus amples informations sur la norme X.400.

III.2 FONCTIONNALITES DU LOGICIEL

L'expéditeur comme le destinataire sont identifiés de manière univoque par un Network User Address (NUA) qui est une adresse de la forme suivante:

S=vincent;OU=ts;O=info;P=fundp;A=rtt;C=be

où C (Country) est le code du pays, puis successivement A (ADMD) pour Administration Management Domain, P (PRMD) pour Private Management Domain, O pour Organization, OU pour Organization Unit et S pour Surname le nom de l'utilisateur.

Les messages créés ou reçus par l'utilisateur peuvent être rangés dans différents dossiers, ainsi on peut organiser son espace de travail tel que les messages soient classés par thèmes, ou par classes de correspondants par exemple.

A un instant donné, un seul dossier est le dossier courant et par défaut, les messages créés y sont rangés. Il est cependant possible de transférer un message d'un dossier à un autre et au sein de chaque dossier un message est identifié par un numéro unique.

Chaque utilisateur possède dans son répertoire, un fichier appelé **profile** qui est créé et initialisé lors de la première utilisation de EAN. Ce fichier que l'on peut modifier par la suite, contient des indications sur la manière dont EAN devra exécuter certaines commandes sans fournir de détails spécifiques.

L'utilisateur peut à sa guise établir une configuration du système telle que certaines commandes soient exécutées d'une certaine manière. Cette configuration peut être temporaire c'est à dire limitée à la durée de la session ou permanente. Dans ce dernier cas, la configuration est sauvée sur le fichier **profile** qui est lu au début de chaque session de EAN.

Dans la plupart des cas, une ligne de commande contient outre le nom de la commande, un argument ou une liste d'arguments séparés par des virgules. Ces arguments servent à altérer le comportement par défaut de la commande.

III.3 ANALYSE DES ECRANS

Dans les copies d'écrans que nous analysons, la chaîne de caractères [EAN]> est l'invite (prompt) du système, telle que nous, en tant qu'utilisateur, l'avions établie dans la configuration de notre fichier profile.

La connaissance des autres options du profile est nécessaire pour interpréter correctement toutes les copies d'écrans que nous présentons. De ce fait, il nous a paru naturel d'analyser en premier lieu l'écran présentant l'ensemble de ces options.

Pour cette analyse, nous supposons que chaque commande efface totalement l'écran avant de présenter les informations résultant de son activation. Nous appellerons dès lors "écran", non pas l'ensemble des informations présentes sur le terminal, mais uniquement les informations concernant la commande (la ligne de commande et les résultats produits).

Avant d'entamer l'analyse individuelle des écrans que nous jugeons représentatifs de l'interface utilisateur, nous formulons d'abord des remarques générales, valables pour tous les écrans. Nous n'aurons ainsi pas à les répéter à chaque analyse.

La première série d'observations concerne le type de dialogue. La version du logiciel que nous avons testé n'utilise que le langage des commandes comme type de dialogue. Conformément aux recommandations ergonomiques, nous avons relevé les manquements suivants:

- l'aide que fournit le logiciel reste très évasive sur la syntaxe des commandes;

- l'usage (intempestif) de signes de ponctuation n'est pas homogène. Certaines lignes de commandes acceptent en effet la virgule comme signe de ponctuation (par exemple `compose cc,bcc,in-reply-to`) et d'autres le point-virgule, notamment lorsque l'argument de la commande est une adresse du type:

`<S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be>.`

- la syntaxe (non homogène) des commandes requiert de la part de l'utilisateur, un effort de mémorisation qui rend l'apprentissage du logiciel lourd.

A propos des entrées des données, il est regrettable que celles-ci ne soient pas toujours suffisamment guidées par des messages d'aide. Une fois la commande `compose` lancée par exemple, l'utilisateur ne peut accéder à aucune aide lui indiquant la signification des champs `cc` et `bcc` par exemple. Il ne saurait non plus comment mettre fin à l'entrée du corps du message.

De plus, le seul dispositif de saisie étant le clavier, les erreurs typographiques et syntaxiques sont fréquentes, surtout dans la phase d'apprentissage du logiciel.

Des remarques générales peuvent aussi être faites à propos du guidage:

- le guidage ne prend malheureusement pas en compte l'existence d'utilisateurs débutants ou occasionnels. En effet, il ne décrit aucun niveau d'explication détaillée, à l'intention des débutants;

- à chaque étape, rien n'indique à l'utilisateur ce qui est attendu de lui. Un novice, qui en début de session, remarque qu'une lettre vient de lui arriver, ne peut savoir comment y répondre. Sans le manuel d'utilisation, il ne pourrait d'ailleurs tout simplement pas adresser le moindre message à un correspondant.

Quant à la présentation des écrans, elle ne tient compte d'aucun critère ergonomique. Toutes les informations sont transmises de manière brute, dans ce sens que la même zone (en l'occurrence tout l'écran) est indistinctement utilisée pour l'entrée des données, l'entrée des commandes et les messages d'erreurs ou de service.

III.3.1 ECRAN D'AFFICHAGE DU FICHIER PROFILE

L'affichage du contenu du fichier profile se fait par la commande **show**. Cette commande admet une liste (éventuellement vide) d'arguments. [4] cite et décrit les différents arguments possibles pour cette commande.

De manière générale, chaque argument permet d'afficher à l'écran des informations concernant une partie bien précise du fichier profile.

Sans arguments, cette commande affiche la totalité du fichier profile.

L'écran E1 ci-dessous illustre ce dernier cas, le lecteur intéressé consultera l'ouvrage de [4] pour les aspects que nous n'abordons pas ici. Dans le cadre de notre travail, nous nous limiterons en effet à une analyse des éléments pour lesquels nous avons des critiques à formuler.

E1: affichage du contenu du fichier profile.

```
[EAN]> show
name          Vincent Kukosama
mailbox       S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be
phone         (02) 358.80.88
alternate     Vincent,V. Kuko, Kukosama
description
address       Av. Charles de Gaulle, 123
              1160 Bruxelles
              Belgique
prompt        [EAN]>
pager         on
printer
tidy          false
list          all
print         from, authorizing-users, to, cc, bcc,
              reply-to, reply-by, in-reply-to,
importance,   sensitivity, subject, obsoletes,
references,   body, status
compose       to, subject
forward       to, subject
reply
auto-edit     false
editor        emacs
file          true
alias         pvb      S=vanbastelaer; OU=ts; O=info;
                      P=fundp; A=rtt; C=be
alias         bdt      detrembleur
alias         pge      Patrick Geurts <S=pge; O=Sunbim;
                      P=uucp; A=rtt; C=be>
alias         jmg      galais
expiry        true
auto-forward
auto-reply
```

La première remarque que l'on peut faire sur cet écran, concerne l'emplacement des informations les plus importantes: les options relatives aux commandes de composition de message, et les noms d'alias notamment.

Celles-ci sont en effet confinées vers le bas de la liste, alors qu'il aurait été plus judicieux, vu leur fréquence d'utilisation, de les grouper en tête et reléguer en queue de liste, les informations telles que l'adresse, le nom et la description de l'utilisateur qui ne sont que très rarement consultées ou mises à jour.

La seconde remarque concerne la présence d'une incohérence dans le vocabulaire. En effet, *printer* signifie bien imprimante mais *print* ne veut pas du tout dire imprimer (à l'imprimante) comme on aurait pu s'y attendre, mais plutôt afficher à l'écran.

III.3.2 ECRANS DE COMPOSITION DE MESSAGE

Les écrans suivants présentent différentes variantes de la commande *compose* dont le but est de composer un brouillon de message. Après sa création, plusieurs actions sont possibles sur le brouillon: l'envoyer, le mettre en attente ou l'archiver. Ce sont respectivement les "*send options*" *send*, *wait* et *file*.

Une remarque globale que l'on peut déjà formuler à propos de ces écrans est qu' il n'y a pas d'homogénéité dans la manière d'indiquer la fin d'une entrée de données.

Pour chaque champ de l'en-tête, la fin est marquée par le passage à la ligne suivante. Pour le corps du message par contre, la fin est indiquée par une autre convention qui en outre n'est pas compatible avec les usages courants: une ligne formée d'un point comme caractère unique en première colonne

Il aurait été préférable que dans tous les cas, la fin d'une entrée de données soit marquée par une touche de validation unique et explicite. Il y aurait eu plus d'homogénéité, avec les avantages qui en découlent.

E2. Commande compose sans argument.

```
[EAN]> compose
To: <S=pge;P=sunbim;A=rtt;C=be>
Subject: tableur lotus

Pourrais-tu m'envoyer la doc lotus ?
.
Send options ? send
[EAN]>
```

L'écran E3 ci-dessous montre que le logiciel reconnaît les paramètres cc, bcc et in-reply-to comme étant des mots clés représentant des champs de l'en-tête du message.

E3. Commande compose avec champs d'en-tête comme arguments.

```
[EAN]> compose cc,bcc,in-reply-to
To: jmg
Subject: Emacs
Cc: bdt
Bcc: pvb
In-reply-to: inbox:2
```

```
La nouvelle version est installée
.
Send options? delete
[EAN]>
```

Sur l'écran E4, bdt et pge qui ne sont pas des mots clés sont (à juste titre) traités comme des noms d'alias.

E4. Commande compose avec alias comme arguments.

```
[EAN]> compose bdt,pge
Subject: MacEan
To: <S=detrembleur;OU=ts;O=info;P=fundp;A=rtt;C=be>
    <S=pge;P=Sunbim;A=rtt;C=be>
```

```
Voici les références du nouveau manuel ....
.
send option? wait
[EAN]>
```

Suite à ces deux écrans, l'utilisateur est en droit de penser que l'écran E5 est valide puisque le logiciel reconnaît les mots clés et traite ceux qui ne le sont pas comme des noms d'alias (ou des adresses). Le résultat est cependant curieux: à cause du nom d'alias en tête, le logiciel ne sait plus ce que cc et bcc veulent dire. Ce comportement est totalement imprévisible car l'aide accessible depuis le logiciel n'aborde pas du tout cet aspect.

Plus grave, cette erreur (car s'en est manifestement une), n'est même pas signalée par le logiciel tant que le message n'est pas envoyé. Pourtant vérifier que cc et bcc ne sont pas des noms d'alias ne poserait aucun problème.

E5. Commande compose avec champs d'en-tête et alias comme arguments.

```
[EAN]> compose bdt,cc,bcc
To: <S=detrembleur;OU=ts;O=info;P=fundp;A=rtt;C=be>
    <S=cc;OU=ts;O=info;P=fundp;A=rtt;C=be>
    <S=bcc;OU=ts;O=info;P=fundp;A=rtt;C=be>
Subject: Retix
.
Send options? delete
[EAN]>
```

L'écran E6 ci-dessous montre la manière dont l'utilisateur peut demander un accusé de remise après la composition et l'envoi du message à son destinataire.

E6. Commande compose avec une demande d'accusé de remise.

```
[EAN]> compose knuts
To: <S=Knuts;OU=ts;O=info;P=fundp;A=rtt;C=be>
Subject: X.400
```

Où en es-tu dans l'exploration
des logiciels X.400 ?

.
Send options? confirm
Message being sent.
[EAN]>

Confirm et report sont des "send options" valides et les lignes de commandes des écrans E4 et E5 montrent que la virgule est un séparateur d'arguments valide. Pourtant sur l'écran E7 ci-dessous, "confirm,report" est considéré comme un seul argument (non valide) et entraîne une erreur.

E7. Commande compose avec deux "send options" séparées par une virgule.

```
[EAN]> compose lamotte
To: <S=lamotte;OU=ts;O=info;P=fundp;A=rtt;C=be>
Subject: X.400
```

Où en es-tu dans l'exploration
des logiciels X.400 ?

.
Send options? confirm,report
Invalid arguments: confirm,report
Send Options? confirm
Message being sent.
[EAN]>

L'écran E8 ci-dessous indique qu'on ne peut avoir qu'un seul brouillon à la fois. Cette restriction n'est fondée sur aucune logique d'utilisation, mais plutôt sur les traitements internes du logiciel, ce qui n'est pas souhaitable si l'on veut rester proche des attentes de l'utilisateur.

E8. Commande compose alors qu'existe déjà un brouillon.

```
[EAN]> compose
There is already a draft message.
[EAN]>
```

La commande compose est en fait désactivée dès qu'un brouillon est créé et le reste tant que celui-ci n'est pas d'une manière ou d'une autre effacé. Aucun signe particulier n'indique à l'écran l'existence d'un brouillon dans le dossier courant. Cette indication aurait diminué la charge informationnelle de l'utilisateur qui ne serait plus obligé de se souvenir qu'un message de brouillon était encore en attente.

L'écran E8 montre aussi que le message de service envoyé par le logiciel n'est pas suffisamment explicite, il se limite à signaler qu'un brouillon existe déjà, sans spécifier l'action à entreprendre pour pouvoir tout de même composer un nouveau message. Faut-il envoyer le brouillon à son destinataire ?, faut-il le détruire ?, l'archiver?, faut-il (ou peut-on) le déplacer dans un autre dossier ? L'utilisateur débutant serait bien embarrassé.

L'écran E9 ci-dessous confirme si besoin en est, le manque d'homogénéité dans les séparateurs d'arguments et ses conséquences sur l'augmentation du risque d'erreur. C'est sur écran en effet qu'on peut se rendre compte que pour les "send options", ce n'est plus la virgule qui est le séparateur valide, mais le blanc.

E9. Commande compose avec deux "send options" séparées par un espace blanc.

```
[EAN]> compose vku
To: <S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be>
Subject: Test auto-reply

Bonjour c'est moi
.
Send options? confirm reports
Invalid arguments: reports
Send options? confirm report
Message being sent.
[EAN]>
```

III.3.3 ECRANS D'AFFICHAGE ET DE RECEPTION DE MESSAGES

E10 est l'écran obtenu suite à la commande `list` dont le but est d'afficher les descriptifs des messages du dossier courant.

Un descriptif de message comprend six champs (colonnes). Le premier indique le numéro d'ordre (chronologique) du message dans le dossier, le second l'état du message. Le troisième indique si le message a été envoyé par l'utilisateur (symbole `->`) ou reçu par lui (absence de symbole). Le quatrième est l'adresse de l'expéditeur (pour les messages reçus) ou du destinataire (pour les messages expédiés), le cinquième et le sixième sont respectivement la date et le sujet du message.

E10. Commande `list` pour l'affichage des descriptifs de messages.

```
[EAN]> list
inbox:
 1  R  ->S=vanbastelaer;Ou=ts;O=i  Oct 13 89 Memoire
 2      S=vanbastelaer;Ou=ts;O=i  Oct 13 89 Memoire
 3      ->S=detrembleur;Ou=ts;O=in Oct 23 89 MacEan
 4      Bernard Detrembleur      Oct 25 89 MacEan
 5  R  ->S=galais;Ou=ts;O=info;P=   Nov 15 89 Emacs
 6  R  ->S=galais;Ou=ts;O=info;P=   Jan 29 90 Emacs II
 7      ->S=pge;O=sunbim;P=uucp;A=   Jan 29 90 Retix
 8      S=MAILER-DAEMON;P=sunbim   Jan 31 90 Returned
mail:
 9      ->S=detrembleur;Ou=ts;O=in   Jan 31 90 essai MacEan
10      ->S=detrembleur;Ou=ts;O=in   Jan 31 90 essai MacEan
11      Bernard Detrembleur        1 11:32 essai MacEan
12  F  S=pge;P=sunbim;A=rtt;C=b     1 11:32 Re: Retix
13      ->S=pge;P=sunbim;A=rtt;C=b   1 12:28 Re: Retix
14      ->S=vanbastelaer;Ou=ts;O=i   1 13:01 retix x.400
15      S=MAILER-DAEMON;P=sunbim     2 10:29 Returned
mail:
16      Jean-Michel Galais          9 15:06 Emacs II
[EAN]>
```

Voici ci-après, les principales remarques que nous retiendrons.

La première a trait à l'absence d'indication du nom du dossier courant avant de lancer la commande, ce n'est qu'a posteriori en effet, que l'on peut se rendre compte que la liste affichée est celle des messages de inbox (dans ce cas-ci).

L'utilisateur doit se souvenir à tout moment, de l'endroit où il se trouve. A défaut il peut évoquer la commande **where** qui indique le nom du dossier courant.

Dans le premier cas un effort supplémentaire, étranger à la nature de la tâche, est demandé à l'utilisateur tandis que dans le second cas la séquence de commandes manque de concision.

La seconde remarque concerne le fait que lorsque la liste à afficher ne peut tenir sur une seule page d'écran, elle ne sera subdivisée en pages d'écran séparées que si l'utilisateur l'a spécifié dans l'option "pager" du profile.

Mais, même dans ce cas, le logiciel n'indique pas automatiquement comment passer à l'écran suivant. En fait, il existe deux possibilités: presser la barre d'espacement affiche la ligne suivante tandis que presser la touche RETURN affiche la page suivante. Aucun guidage ne l'indique.

Pour s'en apercevoir, il faut avant d'actionner la commande list, avoir eu la présence d'esprit de chercher dans l'aide en ligne (commande help), la signification de l'option "pager".

Ce cas illustre une fois de plus la carence totale d'un guidage d'utilisation. Il aurait été plus convivial que d'une part les longues listes d'informations soient toujours subdivisées en pages d'écran séparées, et que d'autre part à chaque pause du défilement, une indication soit donnée sur la manière de redémarrer celui-ci.

La troisième remarque se rapporte à la technique de groupement, ou plus exactement à l'absence de technique de groupement des informations présentées à l'écran.

En effet, les messages sont numérotés par ordre chronologique et apparaissent de manière brute à l'écran. Il est pourtant important pour l'utilisateur de distinguer sans effort, par exemple les messages reçus qu'il a déjà lus et ceux qu'il n'a pas encore lus.

Pour aider l'utilisateur dans l'exploration des informations, cette distinction suggère que les messages lus et ceux qui ne le sont pas soient distinctement groupés dans des zones séparées de l'écran.

La quatrième remarque a trait au fait que les différents champs présentés à l'écran ne portent pas de labels, ce qui rend leur interprétation hasardeuse pour un novice.

A propos du cinquième champ, il faut ajouter que pour les messages 11 à 16, ce n'est pas la date qui est indiquée mais l'heure précédée d'un chiffre. Ce manque d'homogénéité demande un effort supplémentaire pour interpréter correctement ce chiffre comme le jour du mois courant.

Sur l'écran E11 ci-dessous, l'utilisateur a choisi d'afficher le contenu de tous les messages (reçus) qu'il a déjà lus. L'affichage des contenus de messages se fait par la commande **print**.

E11. Commande print pour l'affichage des contenu de messages.

```
[EAN]> print read
Message inbox:2 - Read
From: <S=vanbastelaer;OU=ts;O=info;P=fundp;A=rtt;C=be>
To: Vincent Kukosama
<S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be>
In-Reply-To: pvb:1
Subject: memoire
```

Renseignez-vous au sujet d'interfaces existants pour des softwares x400.

```
Message inbox:4 - Read
From: <S=detrembleur;OU=ts;O=info;P=fundp;A=rtt;C=be>
To: Vincent Kukosama
<S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be>
In-Reply-To: inbox:3
Subject: MacEan
```

OK.

Bernard.

```
Message inbox:8 - Read
From: <S=MAILER-DAEMON;P=sunbim;A=rtt;C=be>
To: Vincent Kukosama
<S=info.fundp.rtt.be!vku;G=ts;O=fun-
cs;P=uucp;A=rtt;C=be>
Subject: Returned mail: Return receipt
```

```
----- Transcript of session follows -----
syslog: send: Network is unreachable
syslog: send: Network is unreachable
syslog: send: Network is unreachable
```

```
----- Message header follows -----
Return-Path: fun-cs!ts.info.fundp.rtt.be!vku
Received: by sunbim.be (4.12/SMI-2.2)
      id AA21240; Mon, 29 Jan 90 14:59:12 -0100
Date: 29 Jan 90 14:04
From: Vincent Kukosama <fun-cs!ts.info.fundp.rtt.be!vku>
To: <pge@ts.info.fundp.rtt.be>
Message-Id: <8:vku@ts.info.fundp.rtt.be>
Subject: Retix
Return-Receipt-To: Vincent Kukosama
      <fun-cs!ts.info.fundp.rtt.be!vku>
```

```
Message inbox:11 - Read
From: <S=detrembleur;OU=ts;O=info;P=fundp;A=rtt;C=be>
To: Vincent Kukosama
<S=vku;OU=ts;O=info;P=fundp;A=rtt;C=be>
In-Reply-To: inbox:9
Subject: Essai MacEan
```

D'accord pour demain matin
[EAN]>

La remarque faite sur l'écran précédent à propos de la subdivision automatique des informations en pages d'écran séparées, reste valable.

Il faut y ajouter plusieurs autres critiques que nous citons ci-dessous.

La plus importante concerne le fait que les messages sont enchaînés en un seul flot d'informations dans lequel il est malaisé de délimiter les frontières de chaque message. Chacun d'eux étant une entité bien distincte, il est essentiel qu'il apparaisse très clairement détaché des autres.

Une autre critique que l'on peut formuler concerne le vocabulaire utilisé pour les messages du système. Des termes tels que "syslog: send: Network is unreachable" sont très techniques et peu expressifs pour l'utilisateur moyen. C'est une illustration du vocabulaire de l'informaticien qui souvent ne reflète pas les usages courants mais plutôt sa culture informatique.

La commande **accept** illustrée ci-dessous par l'écran E12, a pour but d'accepter un message en cours de session. Cette commande est automatiquement activée au début de chaque session.

E12. Commande accept pour la réception de message en cours de session.

```
[EAN]> accept
Accepting messages: .

Reports:
pvb:3:
    Accepted and read by Bernard Knuts
    <S=Knuts;OU=ts;O=info;P=fundp;A=rtt;C=be> on Friday,
    February 16, 1990 at 12:57 GMT
[EAN]>
```

E13. Commande accept pour la réception de message en cours de session.

```
[EAN]> open inbox
[EAN]> accept
Accepting messages: ..

inbox:
  17  NU  Jean-Christophe Lamotte 16 13:00

Reports:
pvb:4:
  Accepted and read by Jean-Christophe Lamotte
  <S=Lamotte;OU=ts;O=info;P=fundp;A=rtt;C=be> on
Friday,
  February 16, 1990 at 12:58 GMT
```

Lorsqu'on observe ces deux derniers écrans de près, on s'aperçoit d'une légère différence en ce qui concerne le contenu du champ "accepting message:". Sur E12 il s'agit d'un point tandis que sur E13 on y voit deux points. S'agit-il du nombre de messages acceptés? le logiciel entretient le plus grand flou à ce propos. En tout cas il y a bien un seul message sur l'écran E12 (un avis de remise généré par le système) et deux sur E13 (un message d'un correspondant et un avis de remise).

Cette information nécessite de la part de l'utilisateur un effort supplémentaire d'interprétation. Il aurait été plus judicieux de dénommer le champ sans ambiguïté, en précisant qu'il s'agit bien du nombre de messages, et que ce nombre soit représenté par un chiffre plutôt que par des points.

En effet, en cas "d'acceptation" de plusieurs dizaines de messages par exemple, il est d'une part très fastidieux de devoir compter les petits points et d'autre part de devoir attendre jusqu'au dernier de ces points avant de pouvoir lire un message.

Remarque.

Les écrans E12 et E13 montrent aussi la manière dont le logiciel envoie vers l'expéditeur, une confirmation de la réception et de la lecture de son message par le destinataire.

En effet, les "reports" qui y figurent sont une illustration des messages du système (par opposition à messages d'utilisateur) auxquels nous avons fait allusion à la section III.1.

III.4 ANALYSE DES ENCHAINEMENTS D'ECRANS

Au chapitre précédent, nous avons supposé qu'un écran, outre la ligne de commande, n'était composé que du résultat produit par la dernière commande. Cette hypothèse n'est en réalité pas vraie car EAN envoie ligne par ligne toutes les informations à l'écran, et sans "rafraîchir" ce dernier entre deux commandes.

Nous avons aussi vu qu'il n'existait qu'une seule fenêtre (tout l'écran) et que celle-ci ne comprenait qu'une seule zone (tout l'écran aussi) où se mélangent indistinctement le dialogue et les messages d'erreurs et de service. Si nous ajoutons à ces carences l'absence de menus, nous aboutissons à la conclusion qu'il n'y a à proprement parler pas d'enchaînement d'écrans.

EAN dans sa version dont nous disposons, ne prévoit aucune commande standard d'enchaînement (retour arrière, sortie annulation, ...) mise à part l'aide par la commande **help**, qui n'est d'ailleurs pas accessible à tout moment (rappelons qu'une fois la commande **compose** lancée, il n'est plus possible en cours de "transaction", d'accéder au "**help**" pour connaître la signification d'un champ de l'entête par exemple).

Il est toutefois possible d'annuler les effets de la commande **compose** grâce à la "send option" **delete**. Les effets des commandes **reply** et **forward** peuvent aussi être annulés de la même manière. Hormis ces cas, EAN n'offre aucune possibilité d'annulation.

III.5 SYNTHÈSE DES CRITIQUES FORMULÉES

Pour clore ce chapitre consacré au diagnostic de l'interface de EAN, nous regroupons systématiquement toutes les critiques formulées, en fonction des paramètres dont tiennent compte les recommandations ergonomiques.

Nous avons mis en lumière les principales carences de l'interface de EAN. Pour être complets, nous devons cependant souligner que le logiciel EAN que nous avons testé n'est pas un produit commercial. C'est plutôt un prototype élaboré par une université: celle de British Columbia à Vancouver au Canada. Par prototype nous voulons dire un logiciel possédant toutes les spécifications du logiciel définitif, mais sans souci de performance ni de convivialité.

Cela explique sans doute la profusion des faiblesses que nous avons relevées.

III.5.1 A PROPOS DU SEQUENCEMENT DES OPERATIONS

- Les séquences des commandes ne sont pas simples, à cause de la syntaxe non homogène. Voir écran E3, E4 et E5;
- il n'existe ni option de retour arrière, ni option de recommencement, ni option de destruction.

III.5.2 A PROPOS DU LANGAGE D'INTERACTION

- Le vocabulaire est parfois très technique et n'est pas approprié à l'utilisateur moyen non informaticien. Voir écran E11;
- il existe des incohérences dans le vocabulaire. Voir écran E11;
- l'utilisation des signes de ponctuation dans les lignes de commande n'est pas homogène. Voir écran E7.

III.5.3 A PROPOS DES ENTREES ET DES DISPOSITIFS D'ENTREE

- Les valeurs par défaut n'apparaissent jamais à l'écran;
- la fin d'une entrée de données n'est pas faite de manière homogène, par une touche explicite. Voir écran E2;
- il n'existe ni label, ni titre, ni message d'aide pour guider l'entrée des données. Voir tous les écrans;
- rien n'indique à l'utilisateur ce qui est attendu de lui à chaque étape;
- le pointage direct (souris ou touches de direction) est inexistant.

III.5.4 A PROPOS DES SORTIES ET DES DISPOSITIFS DE PRESENTATION

- Certaines informations présentées à l'écran nécessitent un effort d'interprétation. Voir écrans E12 et E13;
- pour les données trop longues subdivisées en plusieurs pages d'écran, rien n'est dit sur la manière de passer d'une page à la suivante. Voir écran E10;
- aucune technique particulière de groupement n'est utilisée pour faciliter l'exploration des informations présentées à l'écran. Voir écrans E1 et E11.

III.5.5 A PROPOS DU TRAITEMENT DES ERREURS

- Lorsqu'une erreur est signalée, rien n'est dit sur la manière de la corriger.

III.5.6 A PROPOS DU GUIDAGE

- La description des rôles de chaque commande est insuffisante pour l'utilisateur débutant.

III.5.7 A PROPOS DU TEMPS DE REPONSE

Compte tenu de nos remarques concernant les recommandations ergonomiques sur le temps de réponse, nous n'avons relevé aucune critique sur ce paramètre.

Nous retiendrons cependant que dans le cas de la commande **accept**, si plusieurs messages doivent être "acceptés", l'affichage des points à l'écran entraîne une perte de temps d'autant plus gênante que le nombre de message est grand.

IV. SPECIFICATION D'UNE INTERFACE CONVIVIALE POUR EAN

IV.1 INTRODUCTION

Dans ce chapitre, nous abordons la conception d'une nouvelle interface utilisateur pour le logiciel de courrier électronique EAN.

Nous nous efforcerons de concevoir le dialogue de cette application aussi indépendamment que possible des fonctionnalités sémantiques du logiciel, lesquelles ont été définies par les concepteurs de EAN et pourront être réutilisées telles quelles sous une autre interface: la nôtre.

Nous pensons que d'une part, l'évolution du matériel informatique vers de meilleures performances et capacités et d'autre part, l'évolution des logiciels de gestion d'écran offrant de meilleures possibilités d'interaction entre l'homme et l'ordinateur, doivent nous inciter à concevoir pour EAN, une nouvelle interface plus conviviale que l'originale.

Notre machine cible est une station de travail DEC Station et le logiciel de gestion d'écran dont nous disposons sur cette machine est X-Window, sous le système d'exploitation Unix.

Disposant de ces "facilités" technologiques, nous pensons pouvoir fournir les spécifications nécessaires pour une plus grande souplesse et une plus grande aisance de l'utilisateur dans son dialogue avec l'ordinateur.

Nous visons donc la convivialité envers l'utilisateur mais, en contre partie, nous devons signaler que la tâche du programmeur s'en retrouve plus difficile. X-Window fournit en effet au programmeur un ensemble de primitives fort complexes et relativement difficiles à maîtriser si l'on veut programmer une gestion d'écran très élaborée.

Cette complexité de X-Window induit un long temps d'apprentissage dont nous ne disposons pas. Pour cette raison, nous ne saurions en moins de trois mois, parvenir à réaliser physiquement notre interface.

Dans les sections qui suivent, nous illustrons d'abord à travers MacEAN, un exemple d'interface moderne utilisant les menus déroulants et le pointage direct.

Nous appelons menus déroulants des menus apparaissant en surimpression, à la demande de l'utilisateur et uniquement pendant le temps souhaité par ce dernier.

Après l'illustration de l'interface de MacEAN, nous ferons une brève présentation de X-Window et enfin nous entamerons les spécifications de notre interface.

Dans cette interface, les paramètres constants c'est-à-dire ceux qui ne varient pas d'une application à l'autre (aide à l'utilisation, guidage, aide à la mémorisation), sont traduits en actions qui peuvent être activées par l'utilisateur. Pour chacune d'elles, nous spécifions :

- une dénomination claire, expressive et compatible autant que possible avec des logiciels existants;
- une forme de présentation indiquant si l'action peut être activée ou non;
- un mode de désignation (la souris).

Les paramètres propres à l'application c'est-à-dire propres au courrier électronique, sont traduits en **commandes** activables par l'utilisateur. Pour ces commandes nous spécifions:

- une hiérarchie de l'ensemble;
- une présentation (menus déroulants) et une forme de caractère distinguant les commandes activables de celles qui ne le sont pas;
- le mode de désignation et la syntaxe.

Nous spécifions en outre pour chaque écran, les différentes fenêtres représentant respectivement la zone des messages d'erreurs et de service et la boîte de dialogue qui sert à la fois de zone d'entrée pour les saisies de données et de zone de sortie pour les réponses de l'ordinateur.

Après cette présentation générale, nous spécifions de manière plus formelle la démarche de conception de l'architecture logique. Cette démarche procède en deux étapes: la première organise le travail à accomplir en différents niveaux ordonnés, en vue d'en réduire la complexité et d'avoir un logiciel plus fiable, plus facile à maintenir, plus facile à réutiliser et plus portable. La seconde spécifie la décomposition de chaque couche définie à la première étape, en ses composants ou modules.

IV.2 EXEMPLE DE L'INTERFACE MacEAN

IV.2.1 IDEES GENERALES

Le système Apple MacIntosh permet à l'utilisateur de visualiser les objets qu'il manipule, d'avoir une grande rapidité dans ses actions et d'utiliser la manipulation directe des objets plutôt que le langage de commandes et sa syntaxe souvent complexe.

Ce système est caractérisé par un certain nombre de ressources, dont les principales que gère l'interface sont: les fenêtres, les messages, les icônes et les menus. Ce sont des objets pour lesquels l'interface décrit des caractéristiques telles que: la taille d'une fenêtre, l'emplacement d'une icône et le libellé d'un message.

A ces objets sont associés des événements qui indiquent l'interaction avec l'utilisateur. Parmi tant d'autres, [7] cite principalement les événements suivants:

- presser ou relâcher le bouton de la souris;
- presser ou relâcher une touche du clavier;
- déplacer la souris.

Ces événements permettent principalement de:

- fermer une fenêtre;
- ouvrir une fenêtre;
- modifier la taille d'une fenêtre;
- déplacer une fenêtre;
- faire défiler le contenu d'une fenêtre;
- dérouler un menu;
- activer ou désactiver une commande d'un menu.

Nous avons choisi de présenter comme exemple le système Apple MacIntosh car c'est historiquement le premier système largement diffusé, offrant à l'utilisateur une interface conviviale facilitant l'apprentissage et l'usage du système, grâce à la manipulation de la souris.

Pour de plus amples informations sur le système Apple MacIntosh, nous renvoyons le lecteur aux ouvrages de [20] et [21].

Parmi différents logiciels conçus pour MacIntosh, nous avons retenu MacEAN simplement pour rester dans le domaine du courrier électronique, car rappelons-nous, certaines décisions de conception d'interface dépendent du domaine d'application.

IV.2.2 MACEAN ET LA MANIPULATION DIRECTE

Les concepts de base qui sous-tendent MacEAN sont globalement les mêmes que ceux du logiciel EAN; seules les interfaces diffèrent.

Formellement, selon [17] la manipulation directe correspond à une approche "langage-objet" où des fonctions manipulent des objets. Dans le système MacIntosh, toutes les actions accomplies sur les objets peuvent l'être uniquement avec la souris, l'usage des touches du clavier étant toujours possible bien entendu.

La copie d'écran de la figure IV-1 ci-après montre que MacEAN dans le cadre du menu principal, offre un menu de commandes sur la ligne supérieure, un menu d'actions et de commandes sur la partie latérale droite et dans la zone restante, apparaissent les noms des dossiers et les nombres respectifs de messages qu'ils contiennent.

La commande **Control** permet à l'utilisateur de fixer une configuration particulière, c'est-à-dire d'affecter des valeurs aux paramètres dont le logiciel doit tenir compte lors de l'exécution de certaines commandes (sauvetage automatique ou non de tout message rédigé, définitions des alias, ...).

La commande **Edit** permet d'éditer un message quelconque, la commande **Sheet** permet d'entreprendre diverses actions sur le message de brouillon: l'envoyer, l'éditer, le sauvegarder, ...

La commande **Compose** est utilisée pour rédiger un nouveau message, répondre à un message ou faire suivre un message. La commande **Folders** permet à l'utilisateur d'obtenir la liste de tous ses dossiers (folders).

O	Control	Edit	Sheet	Compose	Folders
===== Folders =====					
Folders	Dest.	# Msg	# New		OK
bdt		5			DESTINATION
inbox	✓	9	1		REMOVE
jmg		3			NEW
vku		5			CANCEL
pvb		4			
pge		2			

Fig. IV-1. Menu principal de MacEAN

La sélection d'un item est matérialisée par l'inversion vidéo (caractères gras sur la figure), ce qui indique à l'opérateur humain l'endroit où doit se focaliser son attention, avant de valider la commande désirée.

Chaque action, comme chaque commande du menu latéral est représentée par un "bouton" étiqueté: OK, DESTINATION, REMOVE, NEW ou CANCEL.

OK et **Cancel** sont des actions, c'est-à-dire des aides à l'utilisation du logiciel. Elles permettent respectivement de valider une commande et d'annuler les effets d'une commande.

Destination, **Remove** et **New** sont des commandes. Nous pensons que c'est par manque de place que les concepteurs de l'interface n'avaient pu les placer sur la première ligne, comme c'est le cas pour toutes les autres commandes.

Destination permet de déterminer le dossier cible où seront rangés tous les messages dont l'utilisateur demande le transfert par la commande **Move**. Le dossier source est le dossier courant.

La commande **New** est utilisée pour créer un nouveau message tandis que la commande **Remove** a pour but la destruction de message.

Ces boutons se comportent comme un menu permanent qui rappelle à l'utilisateur les actions possibles et lui évitent ainsi la mémorisation d'une syntaxe plus complexe.

L'utilisation de la souris pour déplacer le curseur permet d'avoir des actions rapides et simplifie l'apprentissage du logiciel.

Après avoir sélectionné le dossier où il veut travailler, l'utilisateur valide son choix en cliquant sur le bouton OK, ce qui a pour conséquence le passage à l'écran suivant illustré par la figure IV-2. Cet écran donne une image complète du dossier courant (celui sélectionné à l'écran précédent): inbox dans notre cas.

La syntaxe des commandes est postfixée c'est-à-dire que l'objet à traiter est spécifié en premier lieu, suivi par l'opération à accomplir sur cet objet.

===== Messages =====			
Current folder is: inbox There is 9 messages; (1 newly received)			FOLDERS
<div style="display: flex; justify-content: space-between;"> state: outgoing new deleted </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> forwarded unsent report </div>		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 10px; margin: 2px;">OPEN</div> <div style="border: 1px solid black; padding: 2px 10px; margin: 2px;">MOVE</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 2px 10px; margin: 2px;">DELETE</div> <div style="border: 1px solid black; padding: 2px 10px; margin: 2px;">UNDEL</div> </div>	
state	From/To Address	Date	Subject
1 -->	S=jmg;OU=ts;O=info;P=fundp	Oct 12	Emacs
2 N	Bernard de Trembleur	Mar 16	MacEAN
3 NU	Lamotte	Apr 01	X.400

Fig IV-2. MacEAN: image de l'état d'un dossier.

Sur la figure IV-2, du haut vers le bas, la première fenêtre est la barre de titre, la seconde contient deux messages indiquant respectivement le nom du dossier courant et le nombre de messages dans celui-ci. Cette fenêtre comprend en plus la commande **FOLDERS** que l'utilisateur peut activer par cliquage, pour afficher la liste des dossiers.

La troisième fenêtre (à gauche) indique en surbrillance (en caractères gras sur la figure), l'état (new) du message actuellement sélectionné. A sa droite la quatrième fenêtre comprend quatre commandes activables par cliquage aussi: **OPEN**, **MOVE**, **DELETE** et **UNDELETE**, servant respectivement à:

- ouvrir un dossier;
- transférer un message du dossier courant vers un dossier cible préalablement désigné;
- marquer un dossier en vue de sa destruction;
- revenir sur sa décision après avoir marqué un dossier en vue de sa destruction.

La dernière fenêtre présente sur sa première ligne des labels de quatre champs (colonnes): l'état, le NUA, la date et le sujet du message, qui constituent le descriptif du message. Les lignes restantes sont consacrées aux contenus respectifs de chacun de ces champs.

La partie latérale droite de cette dernière fenêtre permet de faire défiler les descriptifs des messages. Il suffit pour cela de placer le curseur sur la petite case supérieure pour un défilement vers le haut et sur la petite case inférieure pour un défilement vers le bas et de cliquer ensuite autant de fois que nécessaire.

L'écran représente virtuellement les objets du monde réel que l'utilisateur peut manipuler. Cette transparence permet à l'opérateur humain d'appliquer son intellect directement sur la tâche qu'il pense, comme si l'outil d'interaction n'existait pas.

A titre d'exemples, les figures suivantes sont obtenues en surimpression à partir de l'écran de la figure IV-1, en activant respectivement les commandes Folders, Compose et Sheet. Les menus correspondants aux commandes Edit et Control sont analogues à ceux illustrés par ces figures. Ces menus déroulants n'apparaissent qu'en réaction à un cliquage. La sélection s'y fait en déplaçant le pointeur sur l'item désiré et en validant par cliquage.

Folders	Compose	Sheet
Hide Msg	Compose Msg	Send Msg
Open Report	Reply to Msg	Send options
Move Msg	Forward Msg	Edit Sheet
Accept New Msg		Save Sheet
Folder Management		.
		.
		.
		Close Sheet

Fig. IV-3. Exemples de menus déroulants de MaceAN.

Pour de plus amples informations sur les Systèmes de Sélection par Menu, nous convions les personnes intéressées à la lecture de l'ouvrage de [5]. Quant aux connaissances sur les relations entre la résolution des problèmes et la perception visuelle, elles peuvent être obtenues dans les ouvrages de [14] et [15] qui affirment entre autres, que les représentations visuelles sont plus faciles à mémoriser et à manipuler que les représentations textuelles ou numériques.

IV.2.3 AVANTAGES DE L'INTERFACE DE MACEAN

L'interface de MaceAN utilise les trois principes ci-après:

- représentation permanente des objets manipulables et des actions possibles sur ces objets;
- utilisation de boutons étiquetés plutôt que d'une syntaxe plus complexe;
- opérations rapides et réversibles dont l'impact sur les objets est immédiatement visible.

De ces principes se dégagent les avantages suivants:

- les novices peuvent apprendre les fonctionnalités de base du logiciel très rapidement (dans le pire des cas, une courte démonstration par un utilisateur expérimenté suffit);
- les messages d'erreurs sont très rarement nécessaires;

- l'utilisateur est soumis à moins d'anxiété car la présentation des écrans est très compréhensible et chaque action est facilement réversible;
- l'utilisateur acquiert une grande confiance et une bonne maîtrise du système car c'est lui qui prend l'initiative et les réponses de la machine sont prévisibles;
- la très courte distance entre la tâche (univers psychologique de l'opérateur humain) et la syntaxe de l'action (univers physique de l'interface) réduit l'effort cognitif requis pour traduire l'intention en actions.

L'interface de MacEAN hérite de toutes les caractéristiques générales des interfaces utilisant des menus comme type de dialogue. La liste exhaustive des avantages qu'offrent ces caractéristiques serait superflue à notre avis, mais nous renvoyons le lecteur qui s'y intéresse aux ouvrages de [18] et [19].

IV.2.4 INCONVENIENTS DE L'INTERFACE DE MACEAN

Les inconvénients que nous citons ci-dessous ne sont pas propres à MacEAN, mais à tout logiciel dont l'interface utilisateur est basée sur la représentation visuelle ou spatiale.

- Une représentation graphique peut avoir une signification claire pour le concepteur, mais pas nécessairement pour l'utilisateur. C'est le cas par exemple de la pomme (représentée par la lettre O) sur le menu des commandes de la figure IV-1;

- pour l'utilisateur expérimenté habitué à l'usage du clavier, "jouer" avec la souris ralentit son travail, surtout s'il est familier avec les abréviations ou autres raccourcis. MacEAN offre aussi la possibilité d'utiliser des combinaisons bien définies de touches du clavier, mais on perd alors la simplicité de la syntaxe et un effort de mémorisation est nécessaire pour bien assimiler toutes les combinaisons de touches;

- la syntaxe postfixée s'adapte mal aux commandes à plusieurs arguments. Pour ces commandes il nous paraît plus naturel en effet, d'indiquer d'abord le nom de la commande, puis ses arguments. Cette syntaxe préfixée traduit mieux à notre avis la manière de penser de l'homme.

A ces inconvénients s'ajoutent des carences ergonomiques propres à l'interface de MacEAN. Nous les citons ci-dessous et présentons pour chacune d'elle la solution que notre interface compte y apporter.

- Il n'y a aucune possibilité de guidage indiquant à chaque étape ce qui est attendu de l'utilisateur.

Pour y remédier, notre interface proposera une zone particulière pour les messages de guidage. Ces messages n'apparaîtront que sur demande de l'utilisateur qui disposera pour cela d'un bouton d'action clairement dénommé.

- Les boutons d'actions ne sont pas séparés des commandes (voir figure IV-1).

Notre interface apportera une plus grande clarté en distinguant nettement: d'une part un menu permanent de commandes et d'autre part une zone ne contenant que des boutons d'actions.

- Aucune aide n'est prévue pour décrire (à l'intention des novices), la syntaxe et le but d'une commande donnée.

Dans notre interface cette lacune sera comblée par la présence d'une action AIDE mise à la disposition de l'utilisateur.

- La commande d'édition **Edit** ne permet de visualiser qu'un seul message à la fois. Cette restriction est gênante, notamment lorsqu'on désire insérer dans l'en-tête du message qu'on édite, le NUA figurant dans l'en-tête d'un autre message.

Pour apporter plus de souplesse lorsqu'on répond à un message, notre interface exploitera les possibilités de multifenêtrage pour visualiser simultanément deux messages dans deux fenêtres différentes.

Nous prévoyons la possibilité de transférer des informations d'une fenêtre à une autre. Cette possibilité est particulièrement utile lorsqu'on veut insérer dans l'en-tête du message qu'on écrit, le NUA se trouvant dans l'en-tête du message auquel on répond.

IV.3 X-WINDOW

UN OUTIL DE MAQUETTAGE D'ECRAN

IV.3.1 INTRODUCTION

Notre but ici n'est évidemment pas de faire du lecteur un fin programmeur d'applications sous X-Window, pareille entreprise dépasse très largement le cadre de notre travail et ne pourrait d'ailleurs être accomplie dans les délais qui nous sont impartis.

Notre objectif est plutôt de présenter dans les grandes lignes, un système d'interaction multifenêtrée sous Unix. Pour cela, nous nous limiterons à discuter brièvement des possibilités mises à la disposition du programmeur d'applications interactives sous X-Window (ou X tout court).

Nous présentons ce logiciel de gestion d'écran car grâce aux possibilités d'interaction qu'il offre, il peut être très utile dans la gestion du dialogue.

Les principaux dispositifs d'entrée supportés par X-Window sont le clavier pour les acquisitions de textes et la souris pour la sélection et le pointage direct.

[11] traite largement de l'ensemble des services de bas niveau et de haut niveau offerts par le système X.

IV.3.2 CONCEPTS GENERAUX

Dans X-Window, une fenêtre représente une section rectangulaire de l'écran. Un processus appelé serveur contrôle tous les dispositifs d'entrée et de sortie; c'est à lui qu'incombe la gestion des fenêtres, de l'affichage de textes et graphiques à l'écran, du clavier et de la souris.

Les différentes structures de données que peut manipuler un programme d'application constituent les ressources du système. Celles-ci comprennent: les fenêtres, les couleurs et les polices de caractères entre autres.

Le programme d'application est appelé *client*, il peut accéder à chaque ressource en formulant une requête contenant l'identifiant de la ressource désirée. La requête est adressée au serveur, lequel offre différents services dont les principaux sont: créer, détruire ou modifier une fenêtre, afficher un texte ou un graphique dans une fenêtre, et afficher des renseignements sur l'état d'une ressource.

IV.3.3 PRINCIPALE RESSOURCE: LA FENETRE

Principale ressource du système, la fenêtre X-Window, ne contient ni barre de titre ni barre de défilement (scroll bar). La barre de défilement d'une fenêtre est une zone rectangulaire de l'écran où il faut placer le curseur, afin que par cliquage, les informations ne pouvant entièrement tenir dans la fenêtre puissent défiler ligne par ligne.

La barre de titre, la barre de défilement et d'autres composants "de haut niveau", peuvent cependant être créés par le programme d'application. Il suffit pour cela de combiner plusieurs fenêtres, comme sur la figure IV-4a où la fenêtre A a la fenêtre C comme barre de titre et la fenêtre E comme barre de défilement.

X organise les fenêtres suivant une hiérarchie ayant une structure d'arbre. Pour chaque écran physique se trouvant sous son contrôle, le serveur X crée automatiquement une fenêtre représentant la racine de l'arbre. Celle-ci occupe la totalité de l'écran et ne peut être déplacée ni modifiée.

Conformément à cette structure d'arbre, chaque fenêtre excepté la racine a un parent ou ancêtre et chaque fenêtre peut avoir une ou plusieurs filles. Sur la figure IV-4a, la fenêtre A, fille de la racine, est l'ancêtre des fenêtres C et E, cette dernière l'étant elle-même de la fenêtre G.

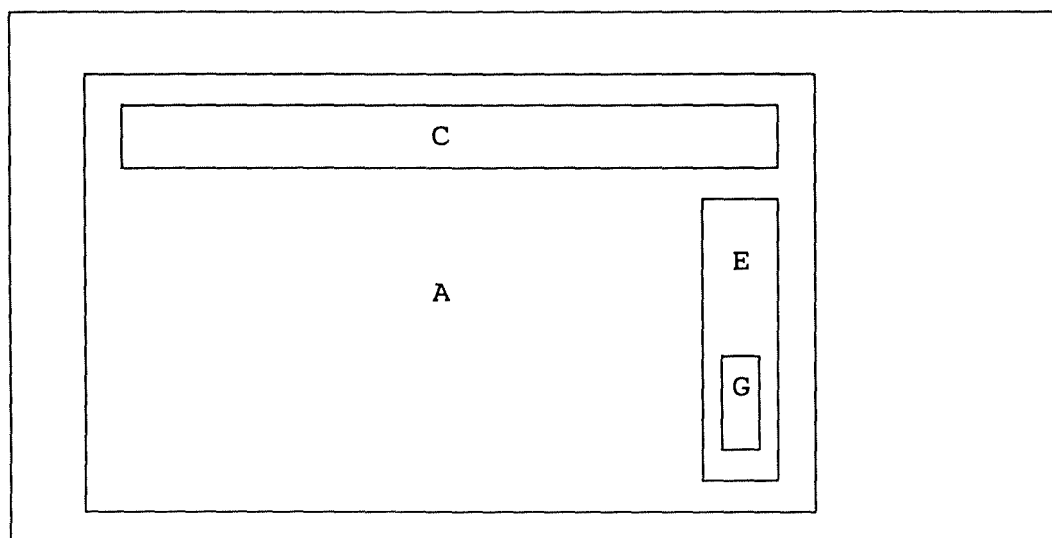


Fig. IV-4a Hiérarchie des fenêtres.

Cette hiérarchie est représentée sous forme d'arbre par la figure IV-4b ci-dessous.

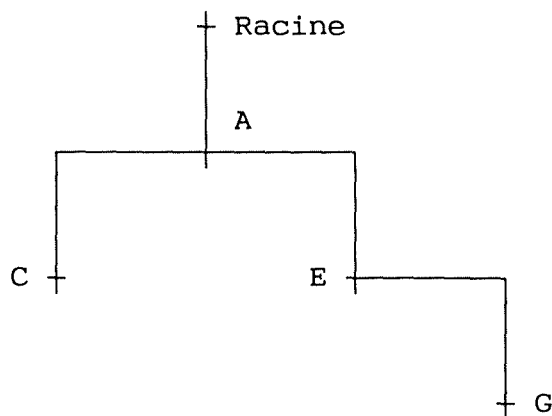


Fig. IV-4b. Structure d'arbre des fenêtres.

Lors de la création d'une fenêtre, le serveur alloue et initialise la structure de donnée correspondant à la fenêtre, mais n'active pas la routine spécifique qui affiche la fenêtre à l'écran. Pour que cet affichage soit effectif, le client doit en plus formuler une requête appelée "map request".

Toutefois, même après une telle requête, une fenêtre peut encore demeurer invisible pour l'une des raisons suivantes:

- la fenêtre est complètement éclipsée par une autre. Elle ne deviendra visible que si celle qui la cache est retirée de l'écran, déplacée, ou si l'ordre d'empilement des deux fenêtres est inversé;

- l'ancêtre de la fenêtre n'est pas visible. La fenêtre ne deviendra visible que lorsque son ancêtre le sera;

- la fenêtre est entièrement située en dehors des limites visibles de son ancêtre. Elle deviendra visible si les dimensions de l'ancêtre sont modifiées de manière telle que la fenêtre soit incluse dans la région visible.

IV.3.4 DIFFERENCES ENTRE X-WINDOW ET LE SYSTEME APPLE MACINTOSH

Contrairement à X-Window, le système Apple MacIntosh n'est pas régi par le modèle client-serveur et ne connaît donc pas la notion de requête adressée au serveur. On parle plutôt d'événement.

Tandis que dans X-Window le serveur offre des services, dans le système Apple MacIntosh on parle de fonctions qui manipulent des objets. Si les ressources dans les deux systèmes sont pratiquement les mêmes, en ce qui concerne plus particulièrement les fenêtres, X-Window les organise en structure d'arbre tandis que cette notion n'est pas explicitement gérée par MacIntosh.

La différence fondamentale réside toutefois dans le fait que la fenêtre X-Window n'est rien d'autre qu'une zone rectangulaire délimitée par un trait continu, alors que la fenêtre MacIntosh est un objet plus complexe et mieux élaboré. Elle peut en effet être définie directement avec une barre de titre, une icône de fermeture et une barre de défilement. X-Window ne connaît pas ces concepts "de haut niveau", mais il est possible de les simuler grâce à la structure d'arbre des fenêtres.

IV.3.5 CONCLUSIONS SUR LE SYSTEME X-WINDOW

X-Window est un système qui peut effectivement aider à l'élaboration d'interfaces très attrayantes pour l'utilisateur. Il existe bien sûr d'autres systèmes de gestion d'écran; notamment TDMS et DecForms dont [12] décrit les principales possibilités.

Cependant de tous ces systèmes, X-Window est le plus complet dans ce sens qu'il offre toutes les possibilités offertes par TDMS et Decforms, mais en plus il possède des fonctionnalités que les autres n'ont pas. Citons entre autres:

- l'usage de la souris comme dispositif d'interaction;
- la possibilité de transférer des informations d'une fenêtre à l'autre;
- la possibilité de créer une barre de défilement;
- la possibilité de déplacer et/ou de modifier les dimensions d'une fenêtre.

Ces possibilités et bien d'autres encore que le lecteur trouvera dans l'ouvrage de [11], sont nécessaires au programmeur pour mettre en oeuvre l'interface que nous souhaiterions.

Il est cependant dommage que son apprentissage soit long et fastidieux. De ce fait, alors qu'il aurait dû décharger le programmeur d'une partie de la mise en oeuvre de l'interface, il lui ajoute au contraire des problèmes de programmation bien plus complexes si ce programmeur est encore débutant dans l'utilisation du logiciel.

Nous sommes convaincus que d'une part, seul un degré d'expertise élevé de la part du programmeur peut lui permettre d'exploiter pleinement X-Window et que d'autre part, l'acquisition d'un tel degré d'expertise n'est possible que si on se consacre pleinement à l'apprentissage et l'usage du logiciel durant une année académique entière. Ne disposant pas de ce temps, nous ne pensons pas pouvoir arriver à l'usage effectif de X-Window pour réaliser notre interface.

IV.4 PRESENTATION GENERALE DES ECRANS

IV.4.1 HIERARCHIE ET PRESENTATION DES COMMANDES

Notre interface est basée sur le multifenêtrage; par multifenêtrage nous voulons dire que l'écran est partagé en plusieurs fenêtres dont la taille et l'emplacement de chacune peuvent être déterminés et modifiés par le système ou par l'utilisateur. De plus, entre ces fenêtres existent des liens hiérarchiques tels que parent-fille et soeur-soeur.

Afin d'optimiser la saisie visuelle des informations présentées à l'écran, nous utilisons des menus déroulants.

Pour cela nous structurons les commandes en deux niveaux. Celles du premier niveau sont affichées en permanence dans une barre de menu appelée de ce fait menu permanent. Les commandes du deuxième niveau sont groupées en sous-ensembles qui recouvrent l'ensemble des commandes EAN.

Le critère de groupement utilisé est la fonctionnalité, mais il aurait été possible de grouper les commandes suivant leurs fréquences d'utilisation. Nous avons cependant estimé que ce dernier critère varie en fonction de l'utilisateur ou du type d'utilisateur. Il ne permet donc pas d'avoir une partition de l'ensemble des commandes aussi naturelle que celle obtenue en groupant par fonctionnalité.

Chaque sous-ensemble est associé à une commande du premier niveau et apparaît sous forme de menu déroulant, chaque fois que la commande correspondante du niveau un est activée. Une commande est activée lorsqu'elle s'exécute.

Nous avons voulu cette décomposition hiérarchique à deux niveaux pour fournir à l'utilisateur, même débutant, une présentation des commandes naturelle et compréhensible sans effort de mémorisation. Le menu permanent des commandes de niveau un se comporte comme un aide-mémoire qui décharge l'utilisateur d'une partie de l'effort cognitif qui lui est demandé.

Pour un complément d'information sur la disposition spatiale des données à l'écran, le lecteur consultera l'ouvrage de [13].

Le menu permanent des commandes occupe la première ligne de l'écran tandis que chaque menu déroulant apparaîtra en dessous de la commande de premier niveau qui lui correspond.

Pour sélectionner une commande deux possibilités sont offertes à l'utilisateur: soit la désignation directe à l'aide de la souris, soit la frappe simultanée d'une touche particulière du clavier et d'une touche correspondant à la première ou à une autre lettre du nom de la commande. Une commande sélectionnée puis validée est activée.

La sélection d'une commande de niveau 1 est matérialisée par l'inversion vidéo du nom de la commande sur le menu permanent et par l'affichage du menu déroulant qui lui correspond. La sélection d'une commande de niveau 2 est matérialisée par l'inversion vidéo du nom de la commande sur le menu déroulant et par l'affichage de la boîte de dialogue qui lui correspond.

Lorsque la commande sélectionnée nécessite l'entrée d'arguments, ceux-ci portent des labels à côtés desquels l'utilisateur écrira ses données. Si l'utilisateur valide une commande sans en spécifier les arguments éventuels, deux situations sont alors possibles: soit la commande accepte des arguments par défaut; dans ce cas, ceux-ci sont pris en compte et l'utilisateur en est averti; soit la commande n'accepte aucun argument par défaut, un message d'erreur est alors présenté dans la fenêtre prévue à cet effet.

Pour la touche particulière nous avons retenu la touche CONTROL, dans un souci d'homogénéité avec d'autres logiciels utilisant la même touche pour les mêmes fins (l'éditeur emacs entre autres).

Les différents sous-ensembles que nous avons retenus sont: **Message, Brouillon, Dossier, Contrôle, Edition, et Session**. Ils constituent le menu permanent des commandes illustré par la figure IV-5a. Leur choix se justifie par notre souci de diviser l'ensemble des commandes EAN en sous-ensembles de commandes fonctionnellement similaires.

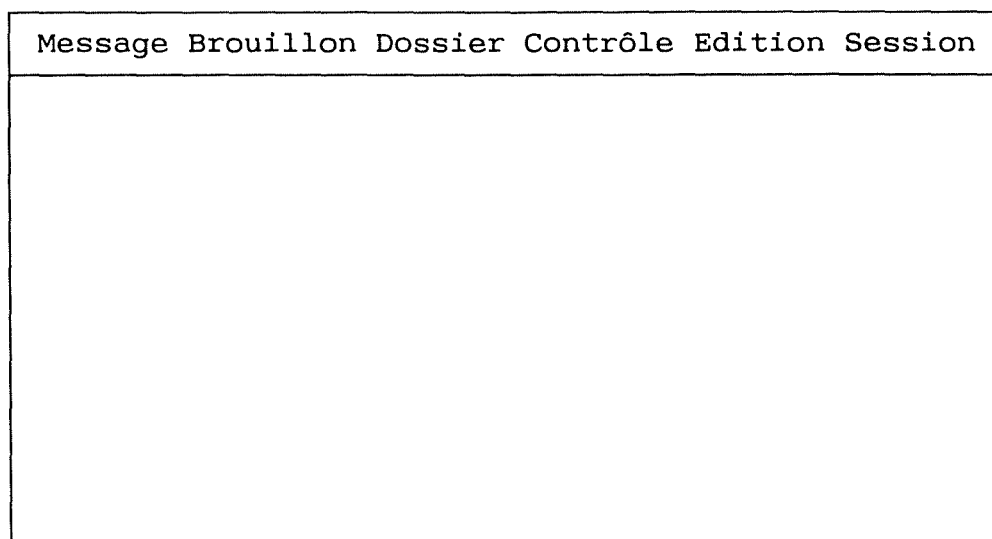


Fig. IV-5a Hiérarchie et présentation des commandes

Cette partition permet à l'utilisateur un meilleur contrôle sur l'ensemble des commandes et érige cet ensemble en une structure arborescente des menus comme indiqué sur les figures IV-5b et IV-5c. Cette structure permet à l'utilisateur d'avoir des actions rapides et d'écourter la phase d'apprentissage du logiciel.

Pour les commandes de premier niveau, la première lettre est identifiante et la frappe de celle-ci ne pose aucun problème pour sélectionner la commande désirée. Quant aux commandes de niveau deux, lorsque plusieurs d'entre elles, dans le même menu, commencent par la même lettre, la sélection se fera en prenant des lettres autres que les premières, de manière à distinguer les commandes. Ces lettres sont alors représentées en majuscules comme sur les figures IV-6 à IV-11.

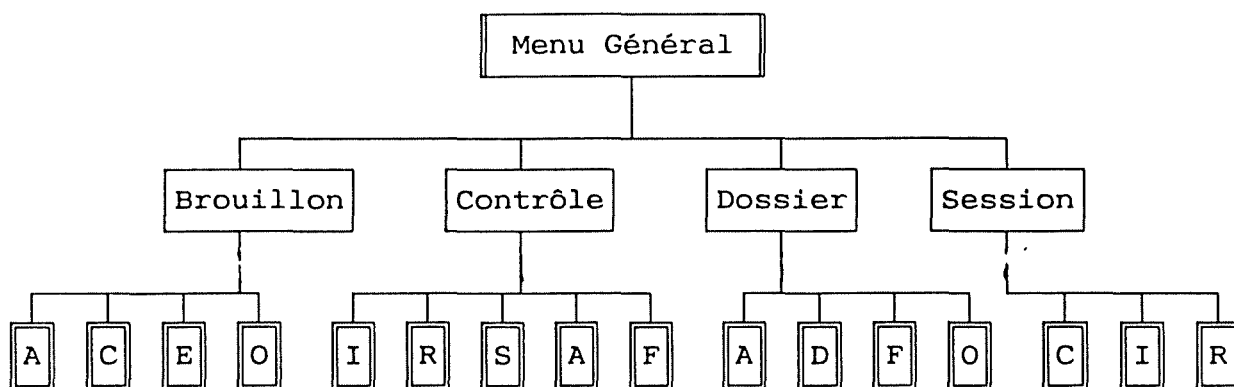


Fig. IV-5b. Arborescence des menus.

La figure IV-5c ci-dessous fait suite à la figure IV-5b sur laquelle nous n'avons pû reprendre toute l'arborescence des menus sur une seule largeur de page.

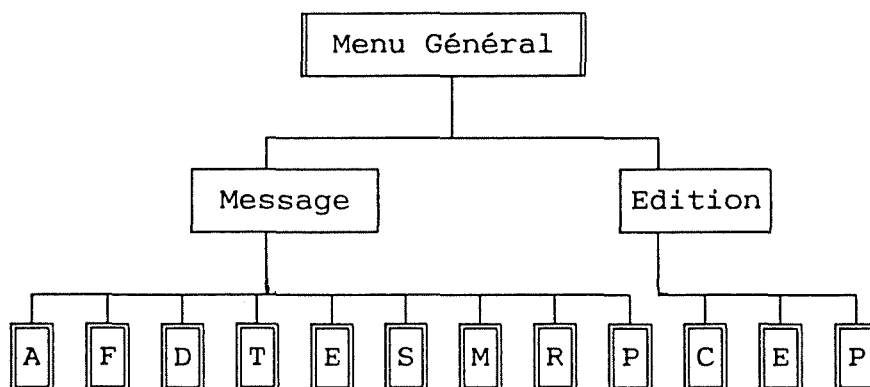

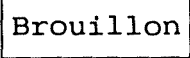
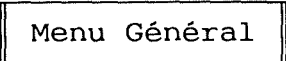


Fig. IV-5c. Arborescence des menus (suite).

Légende des figures IV-5b et IV-5c

 boîte de dialogue affichée après validation de la commande de niveau 2 correspondante (Voir menus déroulants, figures IV-6 à IV-11).

 menu déroulant présenté après validation de la commande de niveau 1 "Brouillon".

 menu permanent des commandes de niveau 1

Nous présentons maintenant les menus déroulants correspondant aux six commandes de niveau 1: Message, Brouillon, Dossier, Contrôle, Edition et Session.

(i) Message.

Lorsque la commande **Message** est sélectionnée, le menu déroulant qui lui correspond présente les commandes suivantes: Accepter, Afficher, Déplacer, Détruire, Faire Suivre, Marquer, Récupérer et Répondre.

Ces commandes de deuxième niveau correspondent respectivement aux commandes EAN Accept, Print, Move, Tidy, Forward, Delete, Undelete et Reply. On trouvera en annexe l'ensemble des commandes EAN citée dans notre ouvrage, chacune accompagnée d'une brève description de son but.

L'écran présentant le menu déroulant correspondant à la commande Message est illustré par la figure IV-6 ci-dessous.

Message
Accepter
aFficher
Déplacer
déTruire
faire Suivre
Marquer
Récupérer
réPondre

Fig. IV-6. Menu déroulant se rapportant à la commande Message.

(ii) Brouillon.

Par la commande **Brouillon**, on accède à toutes les commandes de second niveau manipulant le message de brouillon. Les différentes opérations possibles sur ce dernier sont: Archiver, Créer, Envoyer et Obtenir. Elles correspondent respectivement aux commandes EAN suivantes: File, Compose, Send et Get (voir Annexe). Le menu déroulant correspondant à ces opérations est représenté par la figure IV-7 ci-dessous.

Brouillon
Archiver
Créer
Envoyer
Obtenir

Fig. IV-7. Menu déroulant se rapportant à la commande Brouillon.

(iii) Dossier.

La commande **Dossier** une fois sélectionnée présente au second niveau le sous-ensemble des commandes manipulant les dossiers. Il s'agit des différents choix correspondant aux commandes EAN Folders, Remove, Close et Open (voir annexe).

Notons que nos spécifications ne prévoient pas la possibilité d'activer explicitement la commande **EAN Where**. Celle-ci devient en effet caduque, car le nom du dossier courant sera en permanence affiché à l'écran, sans que l'utilisateur n'ait à le demander.

La fenêtre en surimpression présentant le menu déroulant correspondant est illustrée ci-dessous:

Dossier
Afficher
Détruire
Fermer
Ouvrir

Fig. IV-8. Menu déroulant se rapportant à la commande Dossier.

(iv) Contrôle.

Sous la commande de premier niveau **Contrôle** sont regroupées toutes les commandes de second niveau ayant trait à la configuration de l'environnement de travail de l'utilisateur et à la gestion du directory c'est-à-dire le répertoire des usagers possédant un NUA.

On y retrouve les commandes EAN Register, Find, Drop, Show et Set qui sur le menu correspondent aux commandes Inscrire, Rechercher, Supprimer, Afficheropt (pour Afficher Options) et Fixeroption (pour Fixer Options) comme l'indique la figure IV-9 ci-dessous.

Les commandes Find, Drop et Show ne sont cependant pas disponibles car la norme X.400 sur laquelle est basé EAN, n'offre pas encore le service de directory. Nous les présentons toutefois en vue d'une extension du logiciel, mais nous ne les développerons pas dans la suite.

Contrôle
Inscrire Rechercher Supprimer
Afficheropt Fixeroption

Fig. IV-9. Menu déroulant se rapportant à la commande Contrôle

Bien que l'ensemble des commandes de niveau deux du menu déroulant ci-dessus apparaisse en dessous de la même commande de niveau un (Contrôle), les commandes de niveau deux ne concernent pas toutes la même fonctionnalité. Elles définissent en effet deux fonctionnalités distinctes: la gestion du directory et la configuration du fichier profile.

Ce regroupement de toutes les commandes les moins fréquemment utilisées, en un seul sous-ensemble, est nécessaire pour ne pas surcharger l'écran. La distinction des deux fonctionnalités est toutefois marquée par la séparation de la fenêtre du menu en deux sous-fenêtres.

(v). Edition.

La commande **Edition** permet de modifier par l'utilisation d'un éditeur de texte: l'en-tête du brouillon, le corps du brouillon, ou le fichier profile. C'est ce que représente la figure IV-10 ci-dessous.

Edition
éditer Corps éditer En-tête éditer Profile

Fig. IV-10. Menu déroulant se rapportant à la commande Edition.

(vi). Session.

La commande **Session** illustrée par la figure IV-11 ci-dessous permet de mettre fin (Session-Clôturer) à la session EAN, c'est l'équivalent des commandes EAN Exit, Quit et Stop (voir Annexe) et de suspendre momentanément (Session-Interrompre) la session en sauvegardant l'état courant du système.

Session-Reprendre redémarre une session interrompue, à partir de son point d'interruption. Les deux dernières commandes de niveau 2 ne correspondent à aucune commande EAN.

Session
Clôturer Interrompre Reprendre

Fig. IV-11. Menu déroulant se rapportant à la commande Session.

IV.4.2 BOUTONS D'ACTIONS

Rappelons que les actions correspondent à des aides à l'utilisation et à l'apprentissage du logiciel et doivent permettre à l'utilisateur de contrôler l'enchaînement des opérations.

Les boutons d'actions seront présentés dans une fenêtre prévue à cet effet, située sur la partie droite de l'écran comme l'indique la figure IV-12 ci-dessous. Les différentes actions sont: l'aide, le guidage, la validation, l'action d'annuler et l'action d'interrompre.

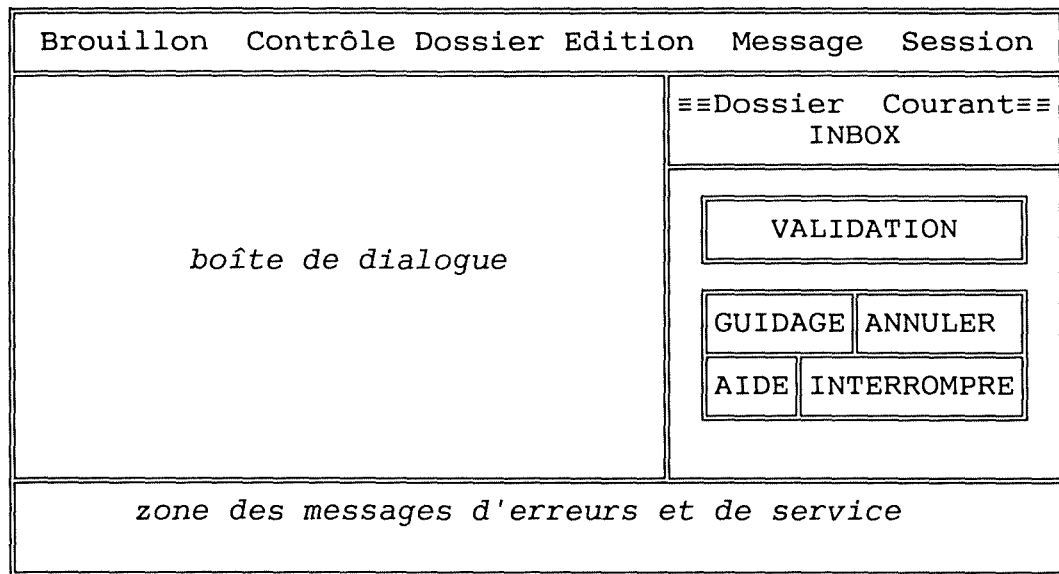


Fig. IV-12. Différentes zones d'un écran.

(i) Aide.

L'action **AIDE** permet à l'utilisateur d'obtenir une aide sur l'utilisation du logiciel c'est-à-dire sur la syntaxe des commandes. Elle correspond à la commande EAN help qui donne la syntaxe et le but d'une commande donnée. L'aide peut être demandée à tout moment.

(ii) Guidage.

Le **GUIDAGE** remplit une fonction qui n'existe pas dans EAN alors qu'elle est essentielle pour un logiciel interactif. Lorsqu'elle est activée, cette action indique à l'utilisateur ce qui est attendu de lui, en fonction de l'endroit où il se trouve dans le logiciel.

Les messages de guidage apparaissent dans une zone de la boîte de dialogue prévue à cet effet. Nous y reviendrons et en donnerons quelques exemples lorsque nous aborderons la section concernant la boîte de dialogue.

(iii) Validation.

La **VALIDATION** est l'action par laquelle l'utilisateur marque explicitement la fin de la saisie des données et confirme l'activation de la commande sélectionnée. Par donnée, nous entendons aussi bien les textes des messages que les arguments des commandes.

Le début d'une transaction est défini comme l'instant où la commande est sélectionnée, tandis que sa fin est marquée par la validation.

L'action de validation est équivalente à la frappe de la touche RETURN du clavier.

(iv) Annuler.

En cours de transaction, l'action d'**ANNULER** permet de revenir à l'état précédent le début de la transaction. Pendant la création d'un message de brouillon par exemple, ANNULER détruit le fragment de message créé jusque là, le dossier courant est inchangé, l'écran principal de la figure IV-12 est de nouveau présenté à l'utilisateur, le menu "Brouillon" est déroulé et le curseur est positionné sur "Créer".

A ce stade, presser de nouveau sur ANNULER remonte d'un niveau dans l'arborescence du menu, ce qui se traduit par l'effacement du menu Brouillon.

A la fin d'une transaction, l'action d'**ANNULER** se comporte différemment selon que la dernière commande validée est réversible ou non. Toutes les commandes sont réversibles, à l'exception de celles permettant de:

- faire suivre, accepter, répondre à un message;
- envoyer le brouillon;
- clôturer la session.

Après validation d'une commande réversible, ANNULER permet encore de remonter l'arborescence du menu, alors que cette remontée n'est plus possible après validation d'une commande irréversible.

(v) *Interrompre.*

L'action d'INTERROMPRE est une aide au travail qui permet à l'utilisateur d'abandonner momentanément la transaction en cours en vue de déclencher une autre opération, avant de revenir à celle abandonnée, à son point d'interruption. Le logiciel doit donc garder une trace de ce point d'interruption .

Hormis la clôture et l'interruption de la session, toute commande est interruptible et cette interruption est matérialisée par l'affichage du menu général c'est-à-dire la racine de l'arborescence du menu. L'utilisateur peut alors déclencher l'opération désirée. Pour revenir à celle abandonnée, il suffit d'activer la commande correspondante.

IV.4.3 BOITE DE DIALOGUE

La boîte de dialogue c'est la zone réservée à l'acquisition et la présentation d'informations, comme l'indique la figure IV-12. Elle est représentée par la fenêtre principale de l'écran. Elle comprend une barre de titre en première ligne, une zone de guidage constituée des dernières lignes et destinée aux messages de guidage et une zone d'entrées-sorties constituée des lignes restantes. Le "titre" de la boîte consiste en la séquence de commandes sélectionnées (commande de niveau 1 - commande de niveau 2).

Dans les sections qui suivent, nous considérons toutes les commandes de premier niveau; à savoir Brouillon, Contrôle, Dossier, Edition, Message et Session. Pour chacune d'elles nous illustrons les différentes présentations de la boîte de dialogue qui correspondent aux commandes de second niveau.

IV.4.3.1 Brouillon.

Les commandes de second niveau illustrées sont: Créer, Archiver, Envoyer et Obtenir.

IV.4.3.1.1 Créer.

La figure IV-13 illustre la boîte de dialogue présentée à l'utilisateur après activation de la commande **Brouillon-Créer**. Pour l'en-tête, seuls les champs spécifiés dans le fichier profile apparaissent. Par défaut comme c'est le cas ici, ces champs sont le destinataire et le sujet du message.

L'écran de la figure IV-13 montre aussi ce que l'utilisateur obtient lorsqu'il demande un guidage: à savoir le message apparaissant dans la fenêtre inférieure.

===== Brouillon - Créer =====
Destinataire: Sujet :
Remplissez l'en-tête, cliquez sur VALIDATION puis entrez le corps du message et cliquez sur VALIDATION.

Fig. IV-13. Contenu de la boîte de dialogue se rapportant à la création d'un brouillon.

IV.4.3.1.2 Archiver.

La figure IV-14 montre le contenu de la boîte de dialogue après activation de la commande **Brouillon-Archiver**. Ici aussi l'action de guidage est illustrée.

==== Brouillon - Archiver ====
Dossier:
Entrez le nom du dossier où sera archivé le message

Fig. IV-14. Contenu de la boîte de dialogue se rapportant à l'archivage d'un brouillon.

IV.4.3.1.3 Envoyer.

La figure IV-15 ci-dessous montre que l'expéditeur peut avant validation de la commande **Brouillon-Envoyer**, indiquer des options d'envoi en cliquant dans les cases respectives réservées à cet effet.

C'est ce que rappelle sur demande de l'utilisateur, le message qui apparaît dans la fenêtre inférieure qui constitue la zone des messages de guidage.

==== Brouillon - Envoyer ====
Archivage 0
Confirmation 0
Avis de remise 0
Avis de non livraison 0
Indiquez les options désirées

Fig. IV-15. Contenu de la boîte de dialogue se rapportant à la commande d'envoi du brouillon.

IV.4.3.1.4 Obtenir.

Après que la commande **Brouillon-Obtenir** ait été activée, l'obtention d'une copie de brouillon d'un message se fait en indiquant le nom du dossier contenant le message et le numéro du message dans ce dossier. C'est ce qu'illustre la figure IV-16 ci-dessous.

===== Brouillon - Obtenir =====
Dossier d'Origine: Numéro du Message:

Fig. IV-16. Contenu de la boîte de dialogue se rapportant à l'obtention d'une copie de brouillon d'un message.

IV.4.3.2 Dossier.

Les commandes de second niveau illustrées sont: Afficher et Détruire. Fermer et Ouvrir ne sont pas illustrées car leurs boîtes de dialogues sont semblables à celle de Détruire.

IV.4.3.2.1 Afficher.

La boîte de dialogue illustrée par la figure IV-17 ci-dessous montre que lorsque l'utilisateur demande l'affichage des contenus des dossiers, grâce à la commande **Dossier-Afficher**, il lui est présenté le nom de chacun des dossiers ainsi que le nombre total de messages et le nombre de nouveaux messages dans chaque dossier.

===== Dossier - Afficher =====		
Nom du Dossier	# Msg	# Nouveaux Msg

Fig. IV-17. Contenu de la boîte de dialogue se rapportant à l'affichage des contenus des dossiers.

IV.4.3.2.2 Détruire (Idem pour Fermer et Ouvrir)

Pour la destruction (ainsi que la fermeture et l'ouverture) d'un dossier, il suffit d'indiquer son nom après activation de la commande **Dossier-Détruire** (idem pour les commandes Dossier-Fermer et Dossier-Ouvrir).

C'est ce qu'illustre la figure IV-18 ci-dessous. Remarquons qu'aucun message explicite de confirmation n'est présenté à l'utilisateur. Un tel message serait superflu puisque la confirmation correspond justement à ce que nous avons appelé validation.

En cliquant sur le bouton de **VALIDATION** l'utilisateur confirme la destruction du dossier tandis que s'il veut annuler la destruction il lui suffit de cliquer sur le bouton **ANNULER**.

===== Dossier - Détruire =====
Nom du Dossier:

Fig. IV-18. Contenu de la boîte de dialogue se rapportant à la destruction d'un dossier.

IV.4.3.3 Message.

Les commandes de second niveau illustrées sont: Accepter, Déplacer, Faire suivre et Répondre. Afficher, Marquer et Récupérer ne sont pas illustrées car leurs boîtes de dialogue sont semblables à celle de Détruire.

IV.4.3.3.2 Accepter.

La commande **Message-Accepter** illustrée par la figure IV-19 ci-dessous permet d'accepter les messages arrivés pendant la session de travail.

L'utilisateur a la possibilité d'accepter soit tous les messages arrivés, soit uniquement ceux d'entre eux qui sont urgents ou importants. Dans ce dernier cas, il a la possibilité de lire d'abord les messages importants, d'y répondre éventuellement et seulement après accepter les messages restants.

Signalons à ce propos que EAN distingue les messages importants de ceux qui ne le sont pas, grâce au champ *importance* de l'en-tête. Le lecteur trouvera dans [4] de plus amples informations sur ce sujet.

===== Message - Accepter =====	
Accepter les messages urgents	0
Accepter tous les messages	0
Cliquez sur la case correspondant à votre choix.	

Fig. IV-19. Contenu de la boîte de dialogue se rapportant à l'acceptation des messages en cours de session.

IV.4.3.3.2 Déplacer.

La commande **Message-Déplacer** illustrée par la figure IV-20 permet de transférer un message ou un groupe de messages d'un dossier vers un autre.

Si le transfert ne concerne qu'un seul message, celui-ci est identifié par son numéro dans le dossier d'origine. Si par contre on veut transférer un groupe de messages, ceux-ci doivent avoir un attribut commun qui les distingue des autres messages ou groupe de messages. De ce fait, cet attribut est appelé **sélecteur**.

Les sélecteurs usuels dans EAN sont: sent, read, deleted, forwarded et new.

- sent permet de sélectionner, dans le dossier courant, tous les messages envoyés par l'utilisateur;
- read permet de sélectionner, dans le dossier courant, tous les messages lus par l'utilisateur;
- deleted permet de sélectionner, dans le dossier courant, tous les messages marqués pour la destruction;
- forwarded permet de sélectionner, dans le dossier courant, tous les messages que l'utilisateur a fait suivre vers d'autres destinataires;
- new permet de sélectionner, dans le dossier courant, tous les messages reçus par l'utilisateur au cours de la session.

===== Message - Déplacer =====
Dossier d'Origine: Numéro du Message:
Sélecteur de Msg :
Dossier Destinataire:
Entrez soit le dossier d'origine plus le N° du msg, soit un sélecteur de message. Entrez ensuite le dossier destinataire.

Fig. IV-20. Contenu de la boîte de dialogue se rapportant au déplacement de message d'un dossier à un autre.

IV.4.3.3.3 Détruire. (Idem pour Afficher, Marquer, Récupérer)

La commande **Message-Détruire**, illustrée par la figure IV-21 ci-dessous, a pour but de détruire un message ou un groupe de messages se trouvant dans un dossier que l'utilisateur identifie par son nom.

Si la destruction ne concerne qu'un seul message, celui-ci est identifié par son numéro dans le dossier, si elle concerne un groupe de messages, ceux-ci sont identifiés par un sélecteur.

===== Message - Détruire =====
Dossier d'Origine: Numéro du Message:
Sélecteur de Msg :
Entrez soit le dossier d'origine plus le N° du msg, soit un sélecteur de message.

Fig. IV-21. Contenu de la boîte de dialogue se rapportant à la destruction de message.

IV.4.3.3.4 Faire Suivre.

Lorsque l'utilisateur reçoit d'un correspondant un message pouvant intéresser une ou plusieurs autres personnes, il a la possibilité de faire suivre ce message vers cette ou ces autres personnes. C'est ce que permet la commande **Message-Faire Suivre** illustrée par la figure IV-22 ci-dessous.

La sélection du message à faire suivre se fait de la même manière que la sélection des messages à déplacer ou à détruire. La désignation du ou des destinataires se fait soit par NUA soit par noms d'alias.

===== Message - Faire Suivre =====
Dossier d'Origine: Numéro du Message:
Sélecteur de Msg :
Destinataire:
Entrez soit le dossier d'origine plus le N° du msg, soit un sélecteur de message. Entrez ensuite la liste des destinataires

Fig. IV-22. Contenu de la boîte de dialogue se rapportant à la commande permettant de faire suivre un message.

IV.4.3.3.5 Répondre.

La commande **Message-Répondre** illustrée par la figure IV-23 permet à l'utilisateur de répondre à un message reçu d'un correspondant.

Outre la fenêtre destinée à l'entrée du message à rédiger, une seconde fenêtre est prévue pour contenir le message auquel on répond.

L'utilisateur peut aisément transférer des informations de la seconde fenêtre vers la première. Cette possibilité est particulièrement utile pour insérer l'adresse du destinataire dans l'en-tête de la lettre qu'on écrit. Il suffit pour cela de recopier le contenu du champ *Expéditeur* de la seconde fenêtre, dans le champ *Destinataire* de la première fenêtre.

La sélection du message auquel on veut répondre se fait en indiquant le dossier contenant le message, ainsi que le numéro du message dans le dossier. Après la désignation du message auquel on veut répondre, celui-ci apparaît dans la seconde fenêtre et son en-tête comprend l'adresse de l'expéditeur et le sujet du message. Nous avons estimé que ces deux éléments sont les plus importants lorsqu'on veut répondre à un message.

Pour sélectionner une portion de texte du message reçu, il suffit à l'aide de la souris, de positionner le curseur au début de la portion désirée, d'enfoncer un bouton de la souris et de le maintenir ainsi pendant qu'on déplace le curseur jusqu'à la fin de la portion de texte. En relachant le bouton de la souris, le bloc sélectionné est présenté en inversion vidéo.

Pour recopier un bloc préalablement sélectionné, il suffit de placer le curseur à l'endroit où l'on désire insérer le bloc et de cliquer simultanément sur deux touches de la souris.

Sur la partie latérale droite de chaque fenêtre, une barre de défilement permet de faire défiler les lignes du message qui n'apparaissent pas dans la fenêtre.

===== Message - Répondre =====	
Réponse	Destinataire: Sujet :
Message reçu Dossier: Numéro :	Expéditeur : Sujet :

Fig. IV-23. Contenu de la boîte de dialogue se rapportant à la réponse à un message.

IV.4.3.4. Contrôle.

Les commandes de second niveau illustrées sont: Afficheropt et Fixeroption. Inscrire, Rechercher et Supprimer ne sont pas traitées pour les raisons que nous avons évoquées au point (iv) du paragraphe IV.4.1.

IV.4.3.4.1 Afficheropt.

La commande **Contrôle-Afficheropt** illustrée par la figure IV-24 ci-dessous permet d'afficher à l'écran les options par défaut ou fixées par l'utilisateur. Ces options dont le logiciel tient compte dans l'exécution de certaines commandes sont contenues dans le fichier profile.

Il s'agit par exemple de l'adresse postale de l'utilisateur, des noms d'alias, des champs d'en-tête qui apparaissent lorsqu'on active la composition de message, de l'éditeur à utiliser lorsqu'on invoque la commande d'édition, d'activer ou non l'éditeur (édition automatique) lorsqu'on invoque la commande de composition de message.

L'utilisateur désigne les options dont il veut afficher les valeurs en sélectionnant par cliquage, les noms de ces options. S'il veut sélectionner toutes les options, il a la possibilité de le faire plus aisément en cliquant sur l'option "toutes".

Une barre de défilement verticale est prévue sur la partie latérale droite. Elle permet de visualiser les options qui n'apparaissent pas à l'écran, suivant le même principe de défilement que nous avons déjà vu plus haut.

===== Contrôle - Afficher Options =====		
Options	Ancienne valeur	Nouvelle valeur
Adresse Alias Cmd Composer Editeur Edition auto		
Toutes		

Fig. IV-24. Contenu de la boîte de dialogue se rapportant à l'affichage des options.

IV.4.3.4.2 Fixeroption.

La commande **Contrôle-Fixeroption** illustrée par la figure IV-25 ci-dessous permet à l'utilisateur de modifier les valeurs des options. La désignation des options dont on veut altérer les valeurs se fait de la même manière que la désignation des options dont on veut afficher les valeurs.

Nous n'avons cependant pas prévu ici la possibilité de désigner globalement toutes les options. Nous avons en effet estimé irréaliste, le fait de devoir modifier toutes les options d'un seul coup.

Comme pour la commande précédente, ici aussi une barre de défilement est prévue.

===== Contrôle - Fixer Options =====		
Options	Valeurs	
Adresse		
Alias		
Cmd Composer		
Editeur		
Edition auto		

Fig. IV-25. Contenu de la boîte de dialogue se rapportant à la modification des options.

IV.4.3.5 Edition.

IV.4.3.5.1 Editer En-tête (idem pour Editer Corps).

La commande **Edition-Editer En-tête** illustrée par la figure IV-26 ci-dessous a pour but de faire appel à l'éditeur que l'utilisateur, par la commande "Contrôle-Fixer Options" aura préalablement spécifié dans l'option "Editeur". Cet appel à l'éditeur permet de modifier l'en-tête du brouillon (dans ce cas ci).

Cet éditeur doit bien entendu exister parmi les utilitaires mis à la disposition de l'usager, sinon un message d'erreur lui signifiera que sa commande ne peut être exécutée.

Après validation, l'édition est alors entièrement sous contrôle de l'éditeur de texte. Nous n'avons prévu aucune facilité supplémentaire à ce niveau-ci car nous pensons que les spécifications de l'interface de l'éditeur sortent du cadre du logiciel EAN.

===== Edition - Editer En-tête =====	
cc	0
bcc	0
To	0
Subject	0
Indiquez les champs de l'en-tête à éditer, en cliquant sur les cases correspondantes.	

Fig. IV-26. Contenu de la boîte de dialogue se rapportant à l'édition de l'en-tête d'un message.

IV.4.4 CRITIQUES ET JUSTIFICATION DE LA PRESENTATION DES ECRANS ET DES DISPOSITIFS D'INTERACTION

Nous nous sommes intéressés au logiciel de courrier électronique EAN et en avons présenté les concepts généraux dans le troisième chapitre. Cette présentation était nécessaire car certaines décisions de conception d'interface dépendent de l'application. C'est notamment le cas du choix de la dénomination des commandes et des libellés des messages d'erreurs.

De ce fait, nous avons cherché à adapter le vocabulaire utilisé dans les diverses présentations d'écrans, au vocabulaire des spécialistes de la tâche, pour autant que celui-ci restait compatible avec le vocabulaire de l'utilisateur moyen.

Pour mieux comprendre les mécanismes qui gouvernent le traitement de l'information par l'homme, particulièrement dans son dialogue avec l'ordinateur, dans le deuxième chapitre nous nous sommes appuyés sur des notions de psychologie cognitive et d'ergonomie.

La psychologie cognitive nous a entre autres appris à distinguer les utilisateurs débutants des utilisateurs expérimentés lors de la conception d'interface. Elle nous a fait prendre conscience des limites de la mémoire à court terme et de l'impact des créations d'automatismes, dans les phases d'apprentissage et d'utilisation d'un logiciel interactif.

Nous nous sommes intéressés à l'ergonomie parce qu'elle étudie l'homme en condition de travail et donc plus particulièrement en condition de travail sur ordinateur. Cette étude porte rappelons-le sur sept paramètres: le séquençement des opérations, le langage d'interaction, les entrées et dispositifs d'entrée, les sorties et dispositifs de présentation, le traitement des erreurs, le guidage et le temps de réponse.

Ce sont ces paramètres que nous analysons ci-dessous pour justifier et critiquer l'interface que nous proposons.

IV.4.4.1 Séquençement des opérations

Nous avons voulu que le séquençement des opérations soit simple. En effet, l'utilisateur peut facilement et rapidement exécuter toute opération en cliquant d'abord sur une commande de niveau 1 et ensuite sur une commande de niveau 2.

Soulignons qu'aucun effort de mémorisation n'est demandé à l'utilisateur puisque un menu permanent de commandes (de niveau 1) lui est présenté. Cette procédure de communication brève et expressive favorise la mise en confiance de l'usager et lui permet d'acquérir rapidement les automatismes nécessaires pour atteindre un bon niveau d'expertise.

Notre interface prévoit d'adapter les séquences des opérations en fonction du degré d'expertise de l'utilisateur. L'entrée des commandes peut en effet se faire par la souris ou par les touches du clavier. L'utilisateur expérimenté a donc la possibilité d'accélérer son rythme de travail en empilant des commandes à partir du clavier tandis que le novice peut grâce à la souris, moduler le séquençement des opérations en fonction de son niveau d'apprentissage.

Empiler des commandes ne peut se faire qu'avec le clavier et consiste à entrer et valider des commandes les unes après les autres, sans attendre que les premières aient produits des résultats. Ainsi on peut par exemple en une seule séquence de commandes empilées, ouvrir un dossier, y détruire tous les messages déjà lus, puis afficher les messages restants.

Relevons que notre interface ne donne pas à l'utilisateur la possibilité de se définir ses propres macro-commandes. Cette possibilité aurait certes été très avantageuse pour les commandes les plus fréquemment utilisées, mais nous avons estimé que le fait de pouvoir sélectionner des commandes par leurs abréviations atténue cette carence.

IV.4.4.2 Langage d'interaction

Le vocabulaire que nous avons choisi pour désigner les commandes s'apparente à celui de l'utilisateur non informaticien. Nous justifions ce choix par le fait que nous estimons que l'apprentissage et l'utilisation du logiciel sont d'autant plus efficaces que les dénominations des commandes sont compatibles avec le vocabulaire de l'utilisateur.

La syntaxe est fort simple puisque grâce au pointage direct sur des menus, elle ne nécessite aucune mémorisation de la part de l'utilisateur. Nous avons aussi veillé à l'homogénéité de la syntaxe en nous assurant que des séquences de commandes identiques conduisent toujours au même résultat. Les risques d'erreurs sont ainsi réduits et d'autre part la réaction de l'utilisateur est plus rapide car elle obéit à des automatismes dus à un environnement constant.

Tenant compte de la disparité des utilisateurs nous avons prévu deux types de dialogues: le langage des commandes permettant d'entrer les commandes à partir du clavier et la sélection des commandes à partir des menus, grâce à l'usage de la souris.

IV.4.4.3 Entrées et dispositifs d'entrée

Nous avons prévu deux dispositifs d'entrée: le clavier et la souris. L'utilisateur peut faire usage du clavier aussi bien pour l'entrée des commandes que pour l'entrée des données, tandis qu'il ne peut utiliser la souris que pour l'entrée des commandes.

Par souci d'homogénéité et de concision, les désignations des commandes à partir du clavier se font par des abréviations de même longueur: une seule lettre chacune. Nous pensons ainsi épargner à l'utilisateur la mémorisation d'informations et de procédures longues et nombreuses.

IV.4.4.4 Sortie et dispositif de présentation

Le dispositif de présentation dont notre interface tient essentiellement compte est l'écran. Nous pensons en effet que dans le dialogue entre l'utilisateur et l'ordinateur, l'écran constitue le principal dispositif de présentation.

En ce qui concerne les sorties sur imprimante, nous suggérerons toutefois l'utilisation d'un papier non ligné, de couleur neutre (blanc ou crème) et de format A4 c'est-à-dire 297 mm de hauteur et 210 mm de largeur. Pour de plus amples renseignements sur la disposition de la lettre sur papier, nous renvoyons le lecteur à l'ouvrage de [23].

La disposition que nous avons adoptée et qui consiste à placer le menu permanent des commandes en première ligne, se justifie par le fait que nous avons cherché à faciliter l'exploration visuelle de l'écran.

En effet, dans son ouvrage [19] montre que lors d'une rapide recherche visuelle d'informations sur écran, la partie supérieure de celui-ci est systématiquement parcourue.

Nous avons structuré l'ensemble des commandes en six groupes, chacun réunissant des commandes fonctionnellement similaires. Nous justifions ce groupement comme suit: sans adopter une structure adéquate, le nombre de commandes est suffisamment grand pour nécessiter de la part de l'utilisateur un effort de mémorisation important.

Pour lui éviter cet effort, nous avons subdivisé la collection des commandes de manière compréhensible (en adoptant un vocabulaire expressif) et naturelle (en groupant suivant la fonctionnalité). Ainsi, dès que l'utilisateur même novice ou occasionnel, sait ce qu'il veut faire, il peut directement pointer la commande désirée sur le menu permanent.

Si son but est par exemple de détruire un message, il sait immédiatement que c'est le menu de la commande **Message** qu'il devra dérouler, sans se soucier des autres commandes apparaissant sur le menu permanent et encore moins des menus déroulants qui y correspondent.

Par ce groupement, nous arrivons donc à réduire le volume d'informations que l'utilisateur doit explorer ainsi que le nombre d'opérations qu'il doit effectuer. La probabilité d'erreur s'en trouve ainsi réduite.

IV.4.4.5 Traitement des erreurs

Notre interface fournit à l'utilisateur aussi bien des moyens de détection que des moyens de correction des erreurs.

Nous avons prévu une zone particulière pour les messages d'erreur, afin d'assurer à l'utilisateur que d'un écran à l'autre, les informations de même nature sont toujours localisées au même endroit. Cette homogénéité permet à l'utilisateur d'être plus sélectif dans la recherche d'informations sur l'écran et améliore donc la détection des erreurs.

De plus, par l'utilisation du pointage direct grâce à la souris, les situations propices à l'erreur sont rares.

IV.4.4.6 Guidage

Nous avons mis à la disposition de l'utilisateur une commande lui permettant à tout instant, en fonction de l'endroit où il se trouve dans l'application, de connaître la liste des opérations possibles.

Nous avons prévu cette facilité surtout pour aider les utilisateurs débutants dans l'apprentissage du logiciel.

IV.4.4.7 Temps de réponse

Notre interface ne gère pas la notion du temps de réponse. Aucun message n'est en effet prévu pour signaler la durée approximative d'une opération en cours. L'utilisateur ne peut donc être en mesure de déterminer si un temps de réponse trop long provient oui ou non d'une erreur de sa part.

Nous justifions ce choix par le fait que nous avons cherché à respecter les recommandations ergonomiques, tout en minimisant la complexité des spécifications de l'interface.

La gestion du temps de réponse, telle que décrite dans les recommandations ergonomiques, nécessiterait en effet une conception d'interface très lourde et occasionnerait des problèmes de programmation que nous avons jugés trop complexes pour être abordés dans le cadre de notre travail.

IV.5 STRUCTURE HIERARCHIQUE DE L'INTERFACE UTILISATEUR

IV.5.1 INTRODUCTION

Nous abordons ici la première étape de la conception de l'architecture logique de l'interface. Cette étape consiste à structurer l'interface de manière hiérarchique.

Structurer un logiciel de manière hiérarchique c'est l'organiser en différents niveaux d'abstraction ordonnés afin d'en réduire la complexité. Cet ordre induit des relations entre les niveaux.

Formellement nous adopterons la définition proposée par [22], définition selon laquelle une structure est hiérarchisée si et seulement si il existe entre ses composants, une relation R qui permet de définir des niveaux tels que:

$$\text{niveau } 0 = \{ A \mid \text{il n'existe pas } B \text{ t.q } R(A,B) \}$$
$$\text{niveau } i = \{ A \mid \text{il existe } B \text{ de niveau } i-1 \text{ t.q } R(A,B) \text{ et } (\text{il existe } C: R(A,C) \Rightarrow C \text{ est de niveau } \leq i-1) \}$$

où $R(A,B)$ signifie qu'il existe une relation de A vers B .

A , B et C sont des composants de la structure hiérarchisée.

La relation que nous considérons est la relation UTILISE, elle induit la hiérarchie UTILISE. Nous disons qu'un composant ou module A UTILISE un composant ou module B , si et seulement si la validité de A dépend de la disponibilité d'une version correcte de B . Par "validité de A " nous voulons dire que A respecte ses spécifications, tandis que par "version correcte de B " nous entendons spécifications correctes de B et conception correcte de B .

IV.5.2 CHOIX DE LA STRUCTURE

Nous envisageons de concevoir l'application interactive en séparant la sémantique du dialogue. Traditionnellement, cette séparation est omise et les fonctions sémantiques utilisent alors le dialogue. C'est cela que [12] appelle "dialogue à l'initiative de la sémantique".

Nous rejetons cette hiérarchisation traditionnelle car elle présente de nombreux désavantages dont le plus flagrant est l'éparpillement des fragments de dialogue au sein de différents modules. L'ensemble du dialogue est alors difficile à modifier et à réutiliser dans d'autres applications.

La structuration que nous adoptons s'inspire d'une part de l'ouvrage de [12] et d'autre part de l'enseignement dispensé par [22].

IV.5.3 JUSTIFICATION DU CHOIX

Ce choix est motivé par le fait que dans leurs travaux, les deux auteurs mettent en exergue l'autonomie du dialogue d'une part et la réutilisabilité et la modificabilité de l'interface d'autre part: points qui correspondent à nos désirs.

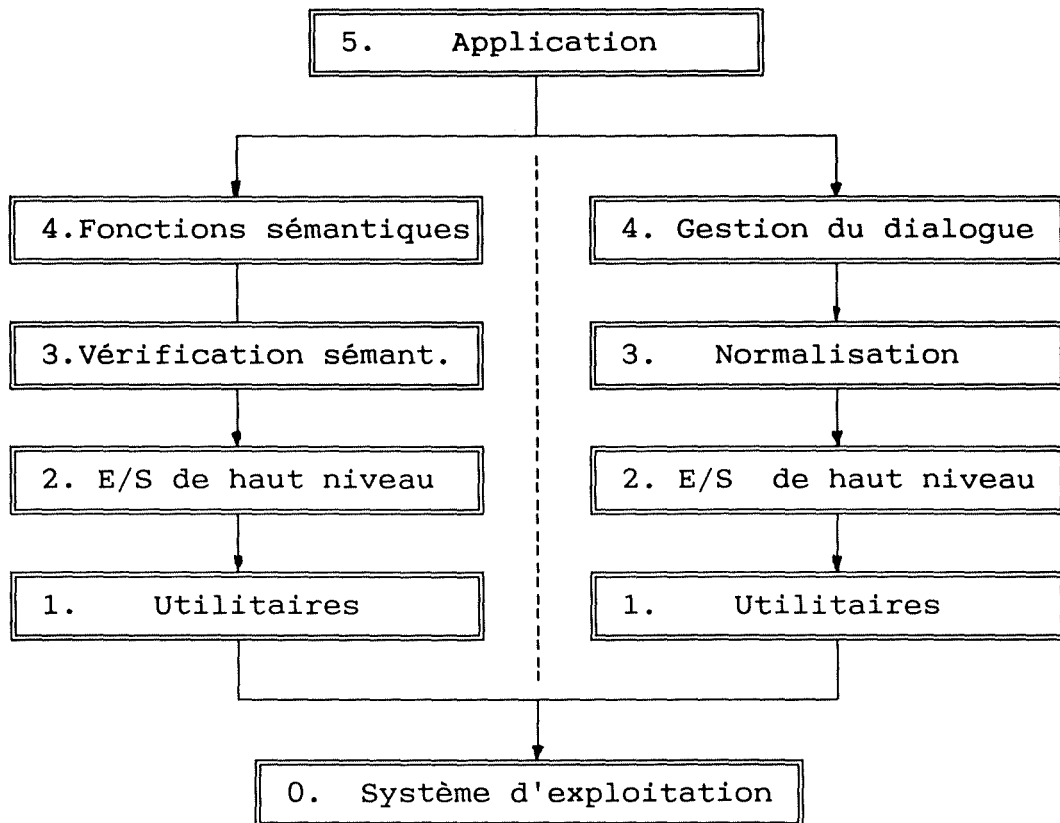
Nous justifions la séparation entre la gestion du dialogue et les fonctions sémantiques par le fait que: nous désirons qu'une décision de conception ou de modification du dialogue, ne puisse affecter la conception des fonctions sémantiques. Dans son ouvrage [12] parle alors d'*autonomie du dialogue*.

Cette autonomie nous permettra de reprendre telle quelle la sémantique originale de EAN, en la "coiffant" de notre interface.

IV.5.4 SCHEMA ET DESCRIPTION DE LA STRUCTURE

La structuration définit six couches ou niveaux d'abstraction comme indiqué sur la figure IV-27 ci-dessous, où la partie gauche représente la sémantique de l'application et la partie droite le dialogue.

Dans le cadre de notre travail, nous ne nous intéresserons qu'à la partie droite de ce schéma c'est-à-dire le dialogue de l'application. La sémantique n'est présentée qu'à titre indicatif, elle ne fait l'objet d'aucun développement.



↓ = UTILISE
 v

Figure IV-27. Différents niveaux d'abstraction. A gauche: la partie sémantique. A droite: la conception de l'interface utilisateur.

Toutes les informations qui doivent passer de la gestion du dialogue vers les fonctions sémantiques et vice-versa, transitent nécessairement par un *coordonateur* situé au plus haut niveau. Ce coordonateur utilise séparément la gestion du dialogue et les fonctions sémantiques.

Nous décrivons dans les paragraphes qui suivent, les fonctionnalités de chacun des six niveaux d'abstraction de l'interface.

Niveau 0: Système d'Exploitation.

L'abstraction faite au niveau du système d'exploitation introduit la notion de *terminal logique* dont l'objectif est d'assurer l'indépendance des couches supérieures vis-à-vis du terminal physique, en leur offrant des services standards tels que positionner le curseur; ces services sont uniformes quelle que soit la particularité du terminal.

Les spécifications de cette couche sortent du cadre de notre travail.

Niveau 1: Utilitaires .

L'objectif de la couche *utilitaires* est de permettre le partage du terminal entre plusieurs applications, ou au sein d'une même application entre plusieurs transactions.

Pour y arriver, l'abstraction faite ici introduit le concept de *terminal virtuel* qui remplace le terminal logique. Les spécifications de ce niveau-ci n'entrent pas non plus dans le cadre de notre travail car nous utiliserons des outils déjà réalisés.

Niveau 2: Entrées/Sorties de Haut Niveau.

La couche *Entrées/Sorties de Haut Niveau* regroupe l'ensemble des services qui permettent de programmer l'acquisition et l'affichage d'informations. On n'effectue plus des entrées/sorties sur des variables séparées, mais sur des structures de données complètes.

Niveau 3: Normalisation.

Les services rendus par la couche *normalisation* sont:

- normaliser les entrées/sorties, c'est-à-dire les débarrasser des erreurs syntaxiques en accordant les facilités d'aide et d'interruption ou annulation de la saisie en cours;
- transformer les données lorsque celles-ci ont une représentation interne (définie par le programmeur) différente de la représentation externe (visible par l'utilisateur).

Niveau 4: Gestion du Dialogue.

La couche *Gestion du Dialogue* offre les services permettant de gérer le contenu et la présentation du message interactif.

L'objectif ici est de rendre la couche supérieure indépendante du type de dialogue (langage des commandes, touches fonctions, menus, question/réponse, ...)

Niveau 5: Application.

On y introduit le coordinateur qui utilise séparément les modules fonctionnels de la sémantique et la gestion du dialogue. Il gère le déclenchement et l'enchaînement des opérations.

IV.5.5 REMARQUES

Nous devons souligner qu'implémenter physiquement une structure qui maintiendrait une séparation stricte entre la partie sémantique représentée par les fonctions sémantiques et la partie lexicale/syntaxique représentée par le dialogue, s'accompagne d'une perte de performance du logiciel. Cette perte de performance est due à un flux important d'informations devant transiter par le coordinateur.

En effet, la vérification syntaxique (par exemple la validité syntaxique d'une commande de destruction de dossier) est séparée de la vérification sémantique (le dossier qu'on veut détruire existe-t-il ?). De ce fait, après la vérification syntaxique par un module de la couche *Normalisation*, ce dernier ne peut directement activer un module de la couche *Vérification sémantique*.

L'ordre d'activation de la vérification sémantique doit remonter jusqu'au coordinateur qui se chargera alors d'activer le module approprié situé dans la partie sémantique.

D'autres solutions sont certes possibles, initialement nous avons entre autres envisagé d'omettre la couche *gestion du dialogue*. Cette solution n'a pas été retenue car elle ne permet pas de concentrer dans un seul module, tous les aspects liés aux libellés, à l'emplacement et à la présentation des messages.

IV.6 STRUCTURE MODULAIRE DE L'INTERFACE UTILISATEUR

IV.6.1 INTRODUCTION

Au cours de la section précédente, nous avons déterminé les différents niveaux d'abstraction de l'interface, nous décomposons maintenant chacun de ces niveaux en ses composants ou modules. La figure IV-28 ci-dessous illustre la manière dont la modularisation que nous adoptons s'intègre dans la hiérarchie UTILISE définie ci-haut.

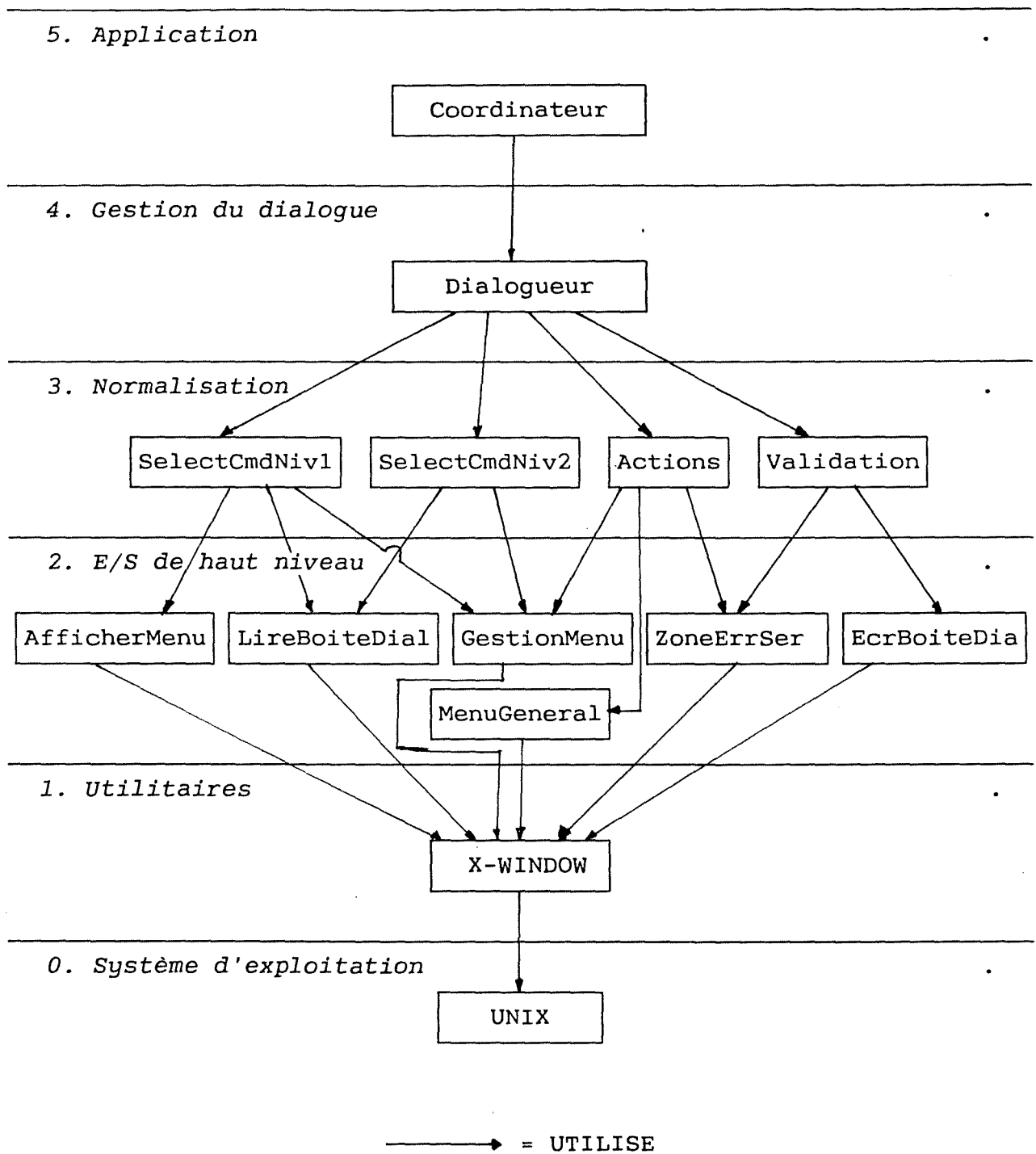


Figure IV-28. Modularisation et hiérarchie "UTILISE".

Dans un premier temps, pour chaque module nous présentons ses spécifications externes, c'est-à-dire l'énoncé de ce que qu'il doit faire sans préciser comment il doit le faire. Ce sont les aspects visibles du module c'est-à-dire les éléments qu'on doit connaître et dont on doit pouvoir faire usage lors de la conception d'autres modules en relation avec lui.

L'étape suivante concernera les aspects invisibles du module, c'est-à-dire ceux qui pour les autres modules ne sont ni connus ni nécessaires à leur conception. Ils sont cependant essentiels pour la conception du module concerné: ce sont les spécifications internes, elles consisteront en la décomposition du module en sous-modules. Seuls les modules nécessitant un tel niveau de décomposition seront spécifiés dans cette étape.

IV.6.2 JUSTIFICATION DE LA MODULARISATION

Dans les paragraphes qui suivent nous justifions niveau par niveau, la modularisation que nous avons adoptée et qui est représentée par la figure IV-28.

(i) Niveau 5.

Dans le coordinateur nous avons rassemblé deux fonctionnalités: la première est le déclenchement et l'enchaînement des opérations et la seconde le passage d'informations entre la partie sémantique et la partie dialogue de l'application.

Nous avons regroupé ces concepts en un seul module car ils permettent d'établir un lien entre la gestion du dialogue et les fonctions sémantiques.

(ii) Niveau 4.

Nous y avons défini un seul module que nous avons appelé DIALOGUEUR. Cela résulte du fait que nous avons cherché à regrouper dans un seul module, tous les aspects liés à la production, la présentation et l'acquisition d'informations à l'écran.

Cette décision de conception assure un faible degré de couplage du module, c'est-à-dire que la dépendance du module avec d'autres est faible, et le nombre de structures de données partagées avec d'autres modules est petit. Cette faible inter-dépendance présente l'avantage de réduire la complexité de la conception du module.

(iii) Niveau 3.

Pour définir les modules de ce niveau d'abstraction, nous avons séparé les traitements qui sont de natures différentes. L'idée est d'isoler des modules réutilisables dans d'autres contextes. Pour cela, nous devons définir les modules de manière telle que la cohésion soit forte au sein de chacun d'entre eux. Par forte cohésion nous entendons une forte inter-dépendance des constituants du module.

De plus, nous avons veillé à ce que le flux d'informations qui passe d'un module à l'autre soit le plus petit possible. Le contraire aurait entraîné une architecture très lourde pour le programmeur.

(iv) Niveau 2.

Les motivations qui sont à la base de la définition des modules de ce niveau d'abstraction sont exactement les mêmes que celles que nous venons de décrire pour le niveau 3.

(v) Niveaux 1 et 0.

Au niveau 1 l'utilitaire essentiel est celui de gestion d'écran. Nous utilisons X-WINDOW et la machine cible est la station de travail DEC Station.

Au niveau 0 le système d'exploitation est Unix, c'est le système sur lequel tourne la machine cible.

IV.6.3 SPECIFICATIONS EXTERNES

Pour chacun des modules de l'interface, nous présentons ci-dessous son objectif, les éléments dont il a besoin en entrée et les éléments qu'il fournit en sortie. Nous décrivons aussi brièvement ses effets, c'est-à-dire les modifications qu'il apporte sur les éléments du logiciel et/ou le flux d'informations qu'il génère.

(i) Module **Coordinateur**.

Objectif: - déclencher et coordonner l'enchaînement des opérations.

Entrée: - le nom valide d'une commande EAN. Ce nom constitue un ordre d'activation d'une fonction sémantique.

Sortie: - le résultat produit par la commande spécifiée en entrée.

Effet: - l'ordre d'activation de la fonction sémantique est exécuté, le résultat produit est retourné au dialogueur.

(ii) Module **Dialogueur**.

Objectifs: - afficher l'écran initial;
- coordonner l'enchaînement des écrans
- transmettre vers le coordinateur, l'ordre d'activation de la commande EAN correspondant à la séquences de commandes sélectionnées.

Entrée: - séquence de commandes sélectionnées.

Sortie: - ligne de commande correcte du point de vue syntaxe EAN. Cette ligne de commande constitue un ordre d'activation de la commande concernée.

Effet: - l'ordre envoyé au coordinateur provoque un flux d'informations provenant des modules fonctionnels. Ces informations sont le résultat produit par l'activation de la commande EAN et sont affichées suivant la présentation de la boîte de dialogue correspondant à la séquence de commandes sélectionnées.

(iii) Module MenuGeneral.

Objectif: - afficher le menu général illustré par la figure IV-12.

Entrée: - néant.

Sortie: - les différentes fenêtres composant l'écran du menu général.

(iv) Module SelectCmdNiv1.

Objectif: - sélectionner et valider un nom de commande de premier niveau et présenter le menu déroulant correspondant.

Entrées: - l'emplacement du menu permanent;
- la liste de noms de commandes de premier niveau.

Sorties: - le nom de la commande de niveau 1 correspondant à la dernière position occupée par le curseur avant la validation;
- l'emplacement du menu déroulant des commandes de niveau 2 correspondant à la commande de niveau 1.

Effets: - le menu des commandes de niveau 2 correspondant à la commande de niveau 1 est déroulé;
- le nom de la commande de niveau 1 est noirci (inversion vidéo).

(v) Module SelectCmdNiv2.

Objectif: - sélectionner et valider un nom de commande de niveau deux et présenter la boîte de dialogue correspondante.

Entrées: - une liste de noms de commandes de niveau 2;
- l'emplacement du menu déroulant correspondant à cette liste.

Sortie: - le nom de la commande de niveau 2 correspondant à la dernière position occupée par le curseur avant la validation.

Effets: - la boîte de dialogue correspondant à la commande de niveau 2 est affichée;
- le nom de la commande de niveau 2 est noirci.

(vi) Module Actions.

Objectif: - fournir les possibilités d'aide à l'utilisation, de guidage, d'annulation et d'interruption de l'activité en cours.

Entrées: - une commande de niveau 1 ou une commande de niveau 1 plus une commande de niveau 2;
- le nom de l'action désirée (Guidage, Aide, Annuler ou Interrompre).

Sorties: - un message dépendant de l'action désirée (voir effets ci-dessous);
- la commande EAN help, si l'action désirée est l'aide.

Effets:

- Si l'action désirée est "Aide" Alors
 - message dans la boîte de dialogue, indiquant la syntaxe et les effets de la commande sélectionnée;
- Si l'action désirée est "Guidage" Alors
 - message dans la boîte de dialogue, indiquant ce qui est attendu de l'utilisateur, compte tenu de l'étape courante de la transaction;
- Si l'action désirée est "Annuler" Alors
 - message de service indiquant que l'état du système d'avant le début de la transaction est restauré,
 - affichage de l'écran d'avant le début de la transaction.
- Si l'action désirée est "Interrompre" Alors
 - message de service indiquant que la transaction est suspendue,
 - affichage du menu général.

(vii) Module Validation.

Objectif: - vérifier puis valider s'il y a lieu, la syntaxe de la commande sélectionnée (commande de niveau 1 et de niveau 2, plus arguments éventuels).

Entrées: - un nom de commande de niveau 1;
- un nom de commande de niveau 2;
- une liste d'arguments éventuellement vide.

Sorties: - un message d'erreur s'il y a une erreur de syntaxe;
- sinon: un message de service et une ligne de commande EAN correspondant à la séquence "commande de niveau 1 - commande de niveau 2". Le message de service doit indiquer que la commande est exécutée ou en cours d'exécution;

Effets: - conversion de la séquence "commande de niveau 1 - commande de niveau 2", en une ligne de commande respectant la syntaxe EAN;
- génération d'un message d'erreur si erreur de syntaxe;
- génération d'un message de service si la commande validée est exécutée ou en cours d'exécution;
- présentation du résultat attendu dans la boîte de dialogue s'il y a lieu.

(viii) Module AfficherMenu.

Objectif: - gérer l'affichage des titres, items et labels dans les différentes fenêtres de menus et dans la boîte de dialogue.

Entrées: - les éléments constituant le texte à afficher;
- les dimensions et la position de la fenêtre où doit apparaître le texte à afficher;
- les positions respectives des éléments dans la fenêtre.

Sortie: - néant.

Effet: - affichage du texte dans la fenêtre, suivant la présentation adéquate.

(ix) Module LireBoiteDial.

Objectif: - gérer l'acquisition des données dans la boîte de dialogue.

Entrées: - les noms des données à lire et leurs positions respectives dans la boîte de dialogue.

Sortie: - néant.

Effet: - les variables représentant les données à lire sont mises à jour.

(x) Module EcrBoiteDial.

Objectif: - gérer la sortie des résultats dans la boîte de dialogue.

Entrée: - le texte résultant de l'activation d'une commande EAN valide.

Sortie: - néant.

Sortie: - le texte entré est présenté suivant le format de la boîte de dialogue correspondant à la commande.

(xi) Module ZoneErrSer

Objectif: - gérer la présentation des messages d'erreurs et de service dans la fenêtre réservée à cet effet.

Entrée: - un texte de message d'erreur ou de service résultant d'une commande EAN.

Sortie: - néant.

Effet: - le texte entré est présenté suivant le format de la zone des messages d'erreurs et de service

(xii) Module **GestionMenu**.

Objectif: - gérer et visualiser le mouvement du curseur d'un item à l'autre dans un menu.

Entrées: - la position et les dimensions de la fenêtre d'un menu;
- la liste des items du menu.

Sortie: - le nom de l'item se trouvant à la dernière position occupée par le curseur avant validation.

Effet: - le nom de l'item se trouvant à la dernière position occupée par le curseur avant validation est noirci.

IV.6.4 SPECIFICATIONS INTERNES

Ci-dessous, nous décomposons en sous-modules les modules de l'interface nécessitant un niveau de raffinement supplémentaire. Cette décomposition vise à mettre en lumière le fonctionnement interne de chacun des modules concernés.

(i) Module **Dialogueur**

i.1 Sous-module **SeparTermArg**

Objectif: - séparer la ligne de commande en terme et arguments de la commande.

Entrée: - une commande de niveau 1, une commande de niveau 2 et des arguments éventuels.

Sortie: - le terme de la commande EAN correspondant à la séquence de commandes entrée;
- la liste des arguments éventuels de la commande.

Effet: - la séquence de commandes entrée est séparée en deux parties: d'une part le terme de la commande et d'autre part les arguments éventuels de la commande.

i.2 Sous-module EnvoiOrdre.

Objectif: - envoyer un ordre d'activation d'une fonction sémantique.

Entrée: - le nom de la commande EAN qui doit être activée;
- liste des arguments dont la commande doit tenir compte.

Sortie: - un message adressé au coordinateur.

Effet: - génération d'un message représentant un ordre d'activation d'une fonction sémantique.

i.3 Sous-module CoordDial.

Objectif: - coordonner le dialogue c'est-à-dire déclencher et gérer l'enchaînement de toutes les procédures définies dans la partie dialogue.

Le déclenchement des ces procédures se fait d'une part, en fonction des commandes sélectionnées et validées par l'utilisateur et d'autre part en fonction des résultats provenant de la partie sémantique de l'application.

Entrée: - une séquence de commandes sélectionnée et validée par l'utilisateur, ou un résultat provenant de l'activation d'une fonction sémantique.

Sortie: - néant.

Effet: - génération d'un ordre d'activation d'un module de la partie dialogue.

(ii) Module Validation.

ii.1 Sous-module ConvertCmd.

Objectif: - convertir les noms de commandes (niveau 1 plus niveau 2) et les arguments sélectionnés, en ligne de commande conforme à la syntaxe EAN.

Entrée: - nom de commande de niveau 1;
- nom de commande de niveau 2;
- liste d'arguments.

Sortie: - soit un message d'erreur ou de service dans la zone appropriée;
 - soit une ligne de commande EAN correcte du point de vue syntaxe.

Effet: - la séquence de commandes sélectionnées et validées, est convertie en ligne de commande EAN.

(iii) Module **GestionMenu**.

iii.1 Sous-module **NoircirItem**.

Objectif: - noircir c'est-à-dire présenter en inversion vidéo, l'item courant (celui où se trouve le curseur).

Entrée: - les coordonnées du curseur.

Sortie: - néant.

Effet: - l'item courant est présenté en inversion vidéo.

iii.2 Sous-module **EteindreItem**.

Objectif: - remettre un item noirci à son état normal (après que le curseur n'y soit plus positionné).

Entrée: - les coordonnées de l'item affiché en inversion vidéo.

Sortie: - néant.

Effet: - l'item est présenté sans inversion vidéo.

iii.3 Sous-module **DeplacerCurseur**.

Objectif: - déplacer le curseur d'un item à un autre.

Entrées: - les coordonnées de l'item où se trouve le curseur avant son déplacement;
 - les coordonnées de l'item suivant.

Sortie: - néant.

Effets: - l'item occupant l'ancienne position du curseur est "éteint";
 - l'item occupant la nouvelle position du curseur est noirci.

iii.4 Sous-module **SelectItem**.

Objectif: - sélectionner le nom de l'item (la commande)
où se trouve actuellement le curseur.

Entrées: - les coordonnées du curseur;
- la liste des noms d'items à partir desquels
la sélection doit s'opérer;
- les coordonnées respectives de ces noms
d'items.

Sortie: - le nom de la commande sélectionnée.

Effet: - la variable représentant le nom de la
commande sélectionnée est mise à jour.

V. CRITIQUES DU TRAVAIL EFFECTUE ET CONCLUSIONS

Nous avons essayé au cours de notre travail de montrer comment réaliser des logiciels interactifs plus adaptés aux attentes de l'utilisateur, c'est-à-dire permettant à ce dernier un apprentissage et une utilisation plus aisés.

Nous avons vu que la conception d'un logiciel interactif pose comme problème majeur l'intégration de trois points de vues: celui de l'ergonome qui apporte une dimension psychologique au problème, celui de l'utilisateur qui pense sa tâche dans un univers qui lui est propre et celui de l'informaticien qui doit concevoir des interfaces plus agréables pour l'utilisateur mais plus difficiles à programmer.

Nous étions en ce qui nous concerne en présence d'un logiciel interactif existant, dont il fallait faire un diagnostic avant de proposer une amélioration de son interface. Comme sources d'informations, nous ne disposions pas du dossier d'analyse mais du logiciel lui-même tel qu'il a été réalisé, ainsi que d'une copie du texte source du programme. Cela nous était suffisant pour pouvoir respectivement; faire des critiques à partir des copies d'écrans que nous avons confrontées aux recommandations ergonomiques et constater que les fonctions sémantiques n'étaient pas séparées du dialogue.

Quant aux spécifications d'une interface plus conviviale, la difficulté principale venait du fait qu'à l'origine, le concepteur de l'application n'avait pas, au vu du code du programme, intégré le point de vue de l'utilisateur dans les spécifications du logiciel. La sémantique n'était pas séparée du dialogue et chaque fonction sémantique pouvait utiliser les modules d'entrée/sortie.

Nous avons tenté de spécifier le dialogue de l'application en le séparant de la sémantique, ce qui nous a conduit à l'obtention des modules logiques structurés, qui constituent l'interface utilisateur que nous proposons.

Ces modules logiques doivent encore être traduits en modules physiques c'est-à-dire en unités codées dans un langage de programmation. Nous aurions voulu arriver jusqu'à ce stade là, mais l'importance de la tâche ne nous l'a pas permis dans les délais impartis.

Notre principal obstacle a été rappelons-le, l'apprentissage de l'utilitaire X-Window. Pour la gestion de l'écran cet utilitaire est indispensable pour mettre en oeuvre les spécifications d'interface que nous avons proposées.

Outre ses nombreuses possibilités, ce logiciel est très portable et permettrait donc à notre interface d'être utilisable sur d'autres machines. En contre partie son handicap vient de la difficulté à maîtriser son utilisation.

De ce fait, nous n'avons pu réaliser concrètement notre interface. Concevoir le programme correspondant à nos spécifications d'interface, tester ce programme, le réajuster, l'intégrer au logiciel, tester cette intégration, l'adapter éventuellement à la sémantique de EAN, documenter le programme définitif, sont les grandes étapes qu'il faut encore parcourir. Nous sommes fermement convaincus qu'à elles seules ces étapes nécessitent une année académique entière.

Le programmeur intéressé peut donc considérer notre travail, notamment le dernier chapitre, comme dossier d'analyse pour mettre en oeuvre l'architecture physique de l'interface

ANNEXE

Dénominations et buts des commandes EAN.

Commande EAN	Dénomination dans notre interface	But de la commande
Accept	Message-Accepter	Accepter la livraison d'un message qui vous a été envoyé.
Close	Dossier-Fermer	Fermer un dossier ouvert préalablement.
Compose	Brouillon-Créer	Créer un brouillon de message.
Delete	Message-Marquer	Marquer un message pour son éventuelle destruction
Drop	Contrôle-Supprimer	Supprimer du service du directotry, toute information à votre sujet.
Edit	Edition-Editer Corps Edition-Editer En-tête Edition-Editer Profile	Invoquer l'éditeur pour éditer le brouillon ou le fichier profile.
Exchange	<i>N'est pas reprise dans notre interface</i>	Modifier la valeur d'un champ de l'en-tête du brouillon.
Exit	Session-Clôturer	Clôturer la session EAN.
File	Brouillon-Archiver	Sauvegarder une copie du brouillon.
Find	Contrôle-Rechercher	Adresser une demande de recherche d'utilisateur au service du directory.
Folders	Dossier-Afficher	Afficher la liste des dossiers.

ANNEXE

Commande EAN	Dénomination dans notre interface	But de la commande
Forward	Message-Faire Suivre	Faire suivre un message reçu d'un correspondant, vers un autre destinataire
Get	Brouillon-Obtenir	Obtenir une copie de brouillon d'un message.
Move	Message-Déplacer	Transférer un message d'un dossier à un autre.
Open	Dossier-Ouvrir	Ouvrir ou créer un dossier selon qu'il existe déjà ou pas.
Print	Message-Afficher	Afficher le contenu d'un message.
Quit	Session-Clôturer	Clôturer la session EAN.
Register	Contrôle-Inscrire	Vous inscrire au service du directory.
Remove	Dossier-Détruire	Détruire un dossier existant.
Reply	Message-Répondre	Répondre à un message reçu d'un correspondant.
Send	Brouillon-Envoyer	Envoyer le brouillon à son ou ses destinataires.
Set	Contrôle-Fixer Options	Fixer les options du fichier profile.
Show	Contrôle-Afficher Options	Afficher les noms et les valeurs des options du fichier profile.
Stop	Session-Clôturer	Clôturer la session EAN.

ANNEXE

Commande EAN	Dénomination dans notre interface	But de la commande
Tidy	Message-Détruire	Détruire effectivement les messages marqués pour la destruction.
Undelete	Message-Récupérer	Effacer les marques des messages marqués pour la destruction.
Where	<i>n'est pas reprise dans notre interface.</i>	Donner le nom du dossier courant.

Références bibliographiques.

- [1] Stephen A. Caswell
"E-MAIL"
Artech House, Boston 1988.
- [2] César Macchi, Jean-François Guilbert
"TELEINFORMATIQUE. Transport et traitement de
l'information dans les réseaux et systèmes
téléinformatiques et télématiques"
Dunod Informatique, Paris.
- [3] Andrew Tanenbaum
"RESEAUX. Architectures, protocoles, applications".
InterEditions, Paris 1990.
- [4] G. Neufeld
"The Ean Distributed Message System User's Manual"
Version 2.1
Distributed Systems Research Group
Department of Computer Science
University of British Columbia, Vancouver
1987
- [5] Ben Shneiderman
"Designing the user Interface structures for
effective human-computer interactives"
Addison Wesley, 1987.
- [6] Dominique Scapin
"Guide ergonomique de construction des interfaces
homme-machine
Rapport INRIA N°77, Le Chesnay 1987. Pages 15 à 48.
- [7] Marie-France Barthet
"Logiciels interactifs et ergonomie."
DUNOD Informatique, Paris 1988.
- [8] Rudolf Arnheim
"Visual Thinking"
University of California Press, Berkeley CA 1972.
- [9] McKim Robert H.
"Experience in Visual Thinking"
Brooks/Cole Publishing Company, Monterey CA 1972.
- [10] Joël Coutaz
"Abstractions for User Interface Design"
IEEE Computer
Sept 1985.

Références bibliographiques.

- [11] Douglas A. Young
"X-Window Systems: programming and applications with Xt"
Prentice Hall, 1989.
- [12] Jean Vanderdonck
"Expérimentation des environnements transactionnels pour les interfaces homme-machine des applications interactives".
Groupe Conception des dialogues FUNDP
Namur, Mars 1990. Pages 6 à 14.
- [13] Christopher F. Herot
"Spatial management of data"
ACM Transactions on Data Base Systems
Volume 5, N° 4, Décembre 1984. Pages 493 à 513.
- [14] Robert V. Rubin, Eric J. Golin and Steven P. Reiss
"Thinkpad: a graphics system for programming by demonstrations".
IEEE Software
Volume 2, N° 2, Mars 1985. Pages 73 à 79.
- [15] Chris Rutowski
"An introduction to the Human Applications Standard Computer Interface, Part 1: Theory and Principles".
BYTE Octobre 1982. Pages 291 à 310.
- [16] Ben Shneiderman
"Control Flow and Data Structure Documentation: Two Experiments".
Communications of the ACM.
Volume 20, 1977. Pages 373 à 381.
- [17] Cranfield Smith, Charles Irby, Ralph Kimball, Bill Verplank and Eric Harslem
"Designing the Star User Interface".
BYTE Avril 1982. Pages 242 à 282.
- [18] James W. Brown
"Controlling the complexity of menu networks".
Communication of the ACM
Volume 20, Juillet 1982. Pages 412 à 418.
- [19] Stuart K. Card
"User perceptual mechanisms in the search of computer commande menus".
Human Factors in computer systems
Mars 1982. Pages 190 à 196

Références bibliographiques.

- [20] Stephen Chernicoff
"MacIntosh Revealed"
Volume 2: programming with the toolbox.
Second edition.
Hayden Books, Indianapolis Indiana, 1987.
- [21] Caroline Rose, Bradley Hacker, Robert Anders
"Inside MacIntosh"
5 volumes.
Addison Wesley, 1989.
- [22] Axel Van Lamsweerde
"Méthodologie de Développement des Logiciels"
Cours dispensé aux étudiants de seconde
licence et maîtrise en informatique des
FUNDP. Namur 1989-1990
- [23] Crispin Michael Geoghegan
"Correspondance Pratique pour tous"
Presses Pocket, Paris 1988.